# International Database Engineered Applications Symposium

*ELTE University*
**August 22-24, 2022**
**Budapest, Hungary**

# IDEAS 2022

**26th**

# International Database Engineered Applications Symposium

The Association for Computing Machinery
2 Penn Plaza, Suite 701
New York, New York 10121-0701

ISBN: 978-1-4503-9709-4

Editors: Will Knight, Gerry Laval
Editorial production by: BytePress/ConfSys
Cover Artwork Logo Copyright BytePress/ConfSys

# Table of Content

# Full Paper

# Full Paper(Continued)

# Full Paper(Continued)

# Short Paper

# Preface

Since its first meeting twenty-six years ago, the IDEAS conference was always–except for the past two years—a place that brought together people who share an interest in data engineering, applications, and science, as well as a passion of sharing that science with like-minded fellow scientists. Fortunately, the Covid pandemic has subsided to the level that Hungary was classified as a safe travel destination by the US Centers for Disease Control and Prevention, and it was possible to meet again in person for most participants, while others participated remotely. The conference also has a tradition of being held in various countries and continents to increase its visibility. This time the conference was held at the ELTE University, in Budapest, Hungary, a site that was chosen by the conference steering committee over a year ago. Since that choice was made, some unforeseen difficulties emerged and posed new challenges, including the tragic war in neighboring Ukraine. The uncertainties led us to schedule the conference for later than usual in the summer, for August 22-24.

The conference covered many topics including big data, block chain, data analytics, machine learning, OLAP, and watermarking. The invited presentations by Prof. András Benczúr, former Dean of Sciences, ELTE University, and Prof. Schahram Dustdar, Director of Distributed Systems Research at TU Wien, were some of the highlights of the conference.

We would like to take this opportunity to thank the members of our program committee, listed here, for their help in the review process. The conference received 38 regular papers and one invited paper that was also reviewed by the program committee. All the submitted papers were assigned to four reviewers, and all program committee members' papers received double blind reviews. The proceedings consist of 1 invited paper, 16 full papers (acceptance rate 42%), and 6 short papers (16%).

This conference would not have been possible without the help and effort of many people and organizations. We would like to express our appreciation to the following people:
-ACM (Anna Lacson, Craig Rodkin, and Barbara Ryan),
-BytePress, ConfSys.org, Concordia University (Will Knight and Gerry Laval),
-ELTE University and the local organization of the conference (Prof. Attila Kiss and Ágnes Kerek),

-Many other people and support staff, who contributed selflessly and have been involved in organizing and holding this event.

We greatly appreciate their efforts and dedication to the conference.

Peter Z. Revesz, Program Committee Chair, IDEAS 2022
Professor, School of Computing, University of Nebraska-Lincoln, USA

Lincoln, Nebraska, August 2022

# Reviewers
# from the Program Committee

* Foto N Afrati(National Technical University of Athens, Greece)

* Ana Sousa Almeida(Instituto Superior de Engenharia do Porto, Portugal)

* Toshiyuki Amagasa(Tsukuba University, Japan)

* Masayoshi Aritsugi(Kumamoto University, Japan)

* Ana Azevedo(Instituto Politecnico do Porto, Portugal)

* Gilbert Babin(HEC Montreal, Canada)

* Christopher Baker(University of New Brunswick, Saint John, Canada)

* Ayse Bener(Ryerson University, Canada)

* Giacomo Bergami(Newcastle University, United Kingdom)

* Jorge Bernardino(Instituto Politecnico de Coimbra, Portugal)

* Christophe Bobineau(Institut National Polytechnique de Grenoble, France)

* Francesco Buccafurri(University of Reggio Calabria, Italy)

* Dumitru Dan Burdescu(University of Craiova, Romania)

* Gregory Butler(Concordia University, Canada)

* Ismael Caballero(Universidad de Castilla La Mancha, Spain)

* Luciano Caroprese(University of Calabria, Italy)

* Rui Chen(Samsung, United States)

* David Chiu(University of Puget Sound, United States)

* Martine Collard(Universite des Antilles, France)

* Carmela Comito(Consiglio Nazionale delle Ricerche, Italy)

* Alfredo Cuzzocrea(University of Calabria, Italy)

* Gabriel David(Universidade do Porto, Portugal)

* Bipin C. Desai(Concordia University, Canada)

* Marcos Aurelio Domingues(Universidade Estadual de Maringa, Brazil)

* Markus Endres(Universitat Augsburg, Germany)

* Nuno Escudeiro(Instituto Politecnico do Porto, Portugal)

* Bettina Fazzinga(Consiglio Nazionale delle Ricerche, Italy)

# Reviewers
# from the Program Committee
# (Continued)

* Alvaro Figueira(Universidade do Porto, Portugal)

* Sergio Flesca(University of Calabria, Italy)

* Alberto Freitas(Universidade do Porto, Portugal)

* Filippo Furfaro(University of Calabria, Italy)

* Benedict Gaster(University of the West of England, Bristol, United Kingdom)

* Sven Groppe(Medizinische Universitat Lubeck, Germany)

* Antonella Guzzo(University of Calabria, Italy)

* Marc Gyssens(Hasselt University, Belgium)

* Irena Holubova(Charles University Prague, Czech Republic)

* Michele Ianni(University of Calabria, Italy)

* Mirjana K Ivanovic(University of Novi Sad, Srebia and Montenegro)

* Nattiya Kanhabua(L3S Research Center, Germany)

* Attila Kiss(Eotvos Lorand University, Hungary)

* Will Knight(ConfSys.org, United States)

* Sotirios Kontogiannis(University of Ioannina, Greece)

* Michal Krátký(Technical University of Ostrava, Czech Republic)

* Georgios Lepouras(University of Peloponnese, Greece)

* Carson K. Leung(University of Manitoba, Canada)

* Chuan-ming Liu(National Taipei University of Technology, Taiwan)

* Grigorios Loukides(King's College London, University of London, United Kingdom)

* Bertil P. Marques(Instituto Superior de Engenharia do Porto, Portugal)

* Elio Masciari(Consiglio Nazionale delle Ricerche, Italy)

* Mirjana Mazuran(POLITECNICO DI MILANO, Italy)

* Giuseppe M. Mazzeo(Facebook, United States)

* Richard Mcclatchey(University of the West of England, Bristol, United Kingdom)

* Peter Mikulecky(University of Hradec Králové, Czech Republic)

# Reviewers
# from the Program Committee
# (Continued)

* Noman Mohammed(University of Manitoba, Canada)

* Yang-sae Moon(Kangwon National University, Korea Republic)

* Kamran Munir(University of the West of England, Bristol, United Kingdom)

* Yiu-kai Dennis Ng(Brigham Young University, United States)

* Mara Nikolaidou(Harokopio University, Greece)

* Selmin Nurcan(Universite Pantheon-Sorbonne (Paris I), France)

* Paulo Jorge Oliveira(Instituto Superior de Engenharia do Porto, Portugal)

* Olga Ormandjieva(Concordia University, Canada)

* Valéria Magalhães Pequeno(Escola Nautica Infante D. Henrique, Portugal)

* Jaroslav Pokorny(Charles University Prague, Czech Republic)

* Giuseppe Polese(University of Salerno, Italy)

* Luboš Popelínský(Masaryk University, Czech Republic)

* Filipe Portela(Universidade do Minho, Portugal)

* Chiara Pulice(University of Calabria, Italy)

* Venkatesh Raghavan(Pivotal Corporation, United States)

* Peter Z. Revesz(University of Nebraska - Lincoln, United States)

* Marina Ribaudo(University of Genoa, Italy)

* Antonino Rullo(University of Calabria, Italy)

* Fereidoon Sadri(University of North Carolina at Greensboro, Reviewer)

* Marinette Savonnet(Universite de Bourgogne, France)

* Younho Seong(North Carolina Agricultural and Technical State University, United States)

* Jianhua Shao(Cardiff University, United Kingdom)

* Atsuhiro Takasu(National Institute of Informatics, Japan)

* Giorgio Terracina(University of Calabria, Italy)

* Stephanie Teufel(University of Fribourg, Switzerland)

* Motomichi Toyama(Keio University, Japan)

# Reviewers
# from the Program Committee
# (Continued)

* Giuseppe Tradigo(University of Calabria, Italy)

* Irina Trubitsyna(University of Calabria, Italy)

* Jeffrey David Ullman(Stanford University, United States)

* Domenico Ursino(Università Politecnica delle Marche, Italy)

* Costas Vassilakis(University of Peloponnese, Greece)

* Krishnamurthy Vidyasankar(Memorial University of Newfoundland, Canada)

* Eugenio Vocaturo(University of Calabria, Italy)

* Ester Zumpano(University of Calabria, Italy)

# External Reviewers

* Samuel Appleby(Newcastle University, United Kingdom)

# IDEAS 2022 - Organization

*Organized by*

**Concordia University, Montreal, Canada;**

**Elite University, Budapest, Hungary**
**with the cooperation of ACM, BytePress and ConfSys**

## Tracks

- **Data Science, Kamran Munir,**
- **GIS Systems and Applications, Bart Kuijpers.**

**Sponsors**
**Bytepress/ConfSys.org**
**Concordia University,**
**In co:operation with ACM**

# Distinguishing Fake and Real News of Twitter Data with the help of Machine Learning Techniques

Aanan Shah
School of Engineering and Computer Science, Laurentian University, Sudbury, Ontario, Canada
ashah2@laurentian.ca

Kalpdrum Passi
School of Engineering and Computer Science, Laurentian University, Sudbury, Ontario, Canada
kpassi@laurentian.ca

## ABSTRACT

News articles have an influence on people's belief and views about various circumstances. In this regard, some news publishers with political or ideological bias try to spread news which are distorted or totally wrong. Natural language processing was used to preprocess the text. Some general features like, number of words, sentences, stopwords, non-alphabetic words, verbs, nouns, and adjectives were identified. Word positioning was labeled to distinguish a word as a noun, a pronoun, an adjective or a verb in the sentences. Preprocessing was followed by feature extraction methods namely, count vectorizer, Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer and word2vec embedding. It was observed that the results obtained by TF-IDF feature extraction method were superior compared with the other two methods. Various machine learning models were used for training the model namely, Naive Bayes, Logistic Regression, Random Forest, K-nearest neighbors (KNN), Support Vector Machine (SVM) and Recurrent Neural Network (RNN) as a deep learning model. The models were successfully tested on two datasets. On the first dataset, SVM achieved an accuracy of 98.5% and RNN achieved an accuracy of 98.03% which is much improvement over the best results of Agarwalla et al., 2019 (83.16 % accuracy). On the second dataset, SVM achieved an accuracy of 97.76%, RNN achieved 97.1% and Logistic Regression achieved 97.50% which is an improvement over the best results of Vijayraghavan et al. 2020 (94.88% accuracy).

## CCS CONCEPTS

• **Computing methodologies** → Artificial intelligence; Search methodologies; Continuous space search.

## KEYWORDS

Natural Language Processing (NLP), Feature extraction, count vectorizer, TF-IDF vectorizer, Word2vec embedding, support vector machine (SVM), Random forest, Naive Bayes, K nearest neighbors (KNN), logistic regression, recurrent neural network (RNN) – LSTM

**ACM Reference Format:**
Aanan Shah and Kalpdrum Passi. 2022. Distinguishing Fake and Real News of Twitter Data with the help of Machine Learning Techniques. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3548785.3548811

## 1 INTRODUCTION

Social media platforms are used to interact with the people where millions of news varieties are uploaded daily and exchanged. These media sites include Facebook, Twitter, WhatsApp, and various others. The social information exchange mediums are somewhat unpredictable and unreliable at spreading the news to the community. Millions of posts are available on the social media sites,, it sometimes becomes highly challenging for the users to predict if the news from the post is real or fake. Above all, the fake news not only affects business growth, it also targets celebrities, politicians, and different famous personalities. This task aims to spread rumors about the targets among the community and manipulate the facts. The situation gets worse when people blindly start to share the news even further. It not only exaggerates the situation but also results in stressful and unwelcoming outcomes.

Along with advancements in the media sector and social networking platforms, the emergence of Artificial Intelligence (AI) in the computing and research sector has revolutionized the world with its enormous applications into various fields of life [1]. The sub-fields of AI, such as machine learning and deep learning, provide the learning algorithms to analyze text and big data. There is a vast majority of such fields, including medical, modeling and simulation, social networking analysis, language processing, graph, audio or video analytics, robotics, or visualization [2]. AI's primary advantage is that it does not require complex modeling and design instead relies on simple equations, but the quality of the data for training and testing the predictive models for the required application must be highly accurate for good results. The fake news detection also has become more interesting with the involvement of AI. The problem of identifying news as a real or a fake belongs to the class of Natural Language Processing (NLP) [3] under the AI domain, The NLP algorithms are used to train the computer to read and decode the human language and extract valuable information out of it. These algorithms are machine learning and deep learning-based solutions to the applications of NLP.

Existing research [4] on fake news detection shows the results using the twitter dataset, but the results are still not as good. This work on fake news detection was undertaken to improve the results on the same dataset and other datasets by implementing various machine learning techniques to increase accuracy. When the accuracy is high, people can easily trust the authenticity of news, which saves the community from fake news. This study has been

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548811

influenced by the work done in [5] and has undertaken some additional techniques to develop a fake news detection mechanism by utilizing deep learning and the NLP approach. Fake News is a report that is deliberately a sham and hoodwinks readers. This tight definition is essential as it can take out the vulnerability between Fake News and other related thoughts, e.g., manufactures and farces [6]. Agarwalla et al. [4] studied to examine how mass media influences the general public's life and how it happens. The dataset was taken from kaggle.com in this research. The dataset dimension is 4008 rows and 4 columns. "URLs", "Function", "Body" and "Name" are the names of the columns. The dataset contains 2136 false news storeys and 1872 genuine news stories. Naïve Bayes classifier gave the maximum accuracy of 83.1% with Lidstone smoothing on the specified preparation set. However, 74% accuracy was achieved in previous models with Naïve Bayes (without Lidstone smoothing) [4]. In those models Logistic Regression was used where the learning rate ($\alpha$) was the critical boundary. The learning rate between 5 and 12 offered the same mixing point, and an approximation of 10 was used. The model brought about exceptionally low accuracy 65.8%. SVM achieved an accuracy of 81.6%.

Looijenga [7] investigates how during the 2012 Dutch parliamentary election campaign, fake messages were used on Twitter. It examines the performance on a Twitter dataset of 8 guided Machine Learning classifiers. The authors claim that with an F1-Score of 88%, the Decision Tree performs the best on the used dataset. Out of 613,033 tweets 328,897 were identified to be real and 284,136 were identified fake tweets. A further 150 tweets have been compiled from the corpus. These messages were being sent by bots or as being fake. Using the Camisani-Calzolari rule set [8], these messages were labeled as bogus. Wynne and Wint [9] propose the Fake news identification framework that considers online news stories' substance. They utilized word n-grams and character n-grams in their investigation. Gradient Boosting accomplished the highest precision of 96% when utilizing character trigram and character four-gram, TF-IDF at 10,000 highlights. Vijayaraghavan et al. [10] have studied fake news detection. In preprocessing, they have removed stop words, punctuation, digits, special characters, and URL links. Then they have compared the distribution of polarity sentiment of the data before and after preprocessing. The texts were tagged to identify the position of the words. By drawing bar plots, they have shown that pronouns were used more in real news. In contrast, adverbs and adjectives were used more in fake news. In the feature extraction step, they have used word2vec embedding, word count vectorizer and TF-IDF vectorizer. They have considered features derived from unigram and bigram words in their word count vectorizer and TF-IDF vectorizer. Before implementing the classification, they have done outlier removal and fine-tuning to get proper tuning parameters for each classification model. In the classification analysis, Artificial Neural Network (ANN) and long-term short memory networks (LSTM), a special case of recurrent neural networks (RNN), were used as deep learning methods. Other classifiers like support vector machines (SVM), random forest (RF), logistic regression (LR) was used. Performing 3-fold cross-validation, they show that the count vectorizer features could get an accuracy of 94.88% in the long-term short memory model (LSTM). The highest accuracy found by word2vec embedding features was derived in ANN as 93.06%. For TF-IDF features, the maximum accuracy was found in logistic regression as 94.79%.

In this study machine learning model for fake news detection was developed using the dataset from kaggle.com [11] the same dataset which was used by (Agarwalla, et al 2019 [4]). The dataset includes labels for fake and real news. Two categories for fake and real news had close proportions, which made the dataset balanced between two categories with 53% fake news and 47% as real news. The results are compared with (Agarwalla et al. 2019) [4]. The accuracy of the SVM model (98%) was much higher compared with the maximum accuracy derived by (Agrawalla et al. 2019) [4] using Naive Bayes classifier with Lidstone smoothing (83%). Since they used only a small number of features (around 100 features) and 1-gram, which considers each feature separately in the document. They also used a threshold of approximately 20% for each selected feature's maximum document frequency. In this research, analysis was performed without using limit on the number of features and using (1,2) gram for the threshold of maximum document frequency. Another dataset from kaggle.com [12] was used to check whether the improvement seen in the first dataset was due to overfitting or whether it is an appropriate approach for increasing the classification performance models in fake news detection. The algorithm was found to be useful in improving the model accuracy for fake news detection in the second dataset as well.

## 2 DATASET DESCRIPTION

In this study, two different types of datasets were used to check the accuracy by training the models and setting the parameters.

### 2.1 First dataset for fake news detection

The dataset contains four columns of URL, Headline news, Body of the news and the class label which show whether the article news is real, or it is fake news [11]. The dataset for machine learning models is appropriate since the class labels are balanced among both groups of fake and real news. The proportion of label categories (real and fake) is not much different from each other, which does not bias a specific class due to different prior probability for one class label. The frequency of each group of the label is shown in Figure 1.

The dataset includes 4009 article news. There are 2137 (53.3%) fake news, and 1872 (46.7%) real news. The URL and Headline of the news are complete for all 4009 articles. But for Body, there are 17 actual news articles and four fake news articles with no or missing Body. There are 12 unique hostnames in this dataset. These news articles are taken from these 12 news websites: "abcnews", "before its news", "bleacher report", "clarivate", "dailybuzzlive", "activist post", "BBC", "CNN", "disclose Tv", "NYTimes" and "Reuters." There are two articles from api.content-ad.net.

### 2.2 Second dataset for fake news detection

This dataset was taken from kaggle.com [12] [13]. The columns in this dataset are the title (headline), author of the news, text (body) and the label, which specify whether it is fake news (1) or real news (0). Some of the news has a missing title, author name or body. This data includes 20,800 news articles. From 20,800 articles, 10,387 (49.9%) are fake news and 10,413 (50.1%) are real news. The dataset

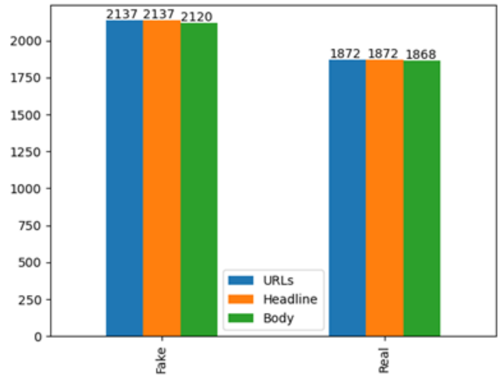


**Figure 1: Frequency of fake and real news in the dataset.**

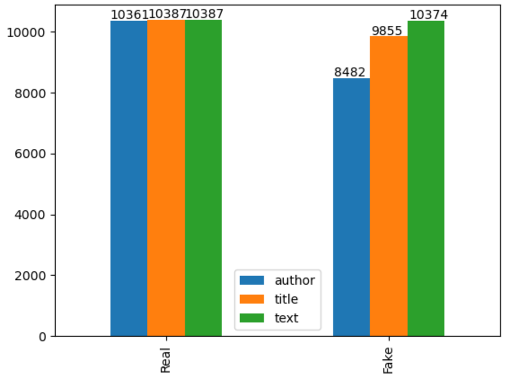


**Figure 2: Frequency of fake and real news in the dataset**

is balanced between the two categories of real and fake news. The frequency plot for author name, title and body for fake and real news is shown in Figure 2.

For author names there are 1957 missing author names (1931 missing in fake news and 26 missing in real news), no missing item for title and text of real news and there are 558 missing titles for fake news and 39 missing in text of the fake news. These 558 + 39 = 597 news were removed from the dataset. After removing them there are 20,203 news where 10,387 (51.4%) of them are real news and 9,816 (48.6%) are fake news. The dataset is still balanced and could be used for classification analysis.

## 3 DATA PREPROCESSING

Preprocessing is the most crucial step in machine learning. Certifiable material is frequently incomplete, temperamental, or otherwise absent. Such propensities or examples are likely to include a few errors. Data pre-processing is a way of addressing these problems. Whatever data we get from Twitter are unfinished, inexact, or it might have some errors, like missing values, null values etc. Before we perform any task in NLP, we must preprocess the data or clean the data to increase the data's quality and make it meaningful and readable. After we process the data, the data's size would be decreased so we can handle it very accurately [13]. In this research, python and its libraries were used to perform preprocessing on the data. Preprocessing will cause all the digits, punctuations, stopwords, URLs to be removed from the news article. Preprocessing includes stopword removal, punctuation and digit removal, URL removal, separation of sentences, word tokenizing, word positions in a sentence, and converting words in lower case, word lemmatization, word tagging and concatenation.

## 4 FEATURE EXTRACTION AND CLASSIFICATION

After cleaning the data, it should be mapped into the numeric presentation in the form of vectors. It is a part of the reduction process of dimensionality. A large dataset contains many variables. Feature extraction helps to get the best features from big datasets to increase the accuracy of the model. Using Feature Extraction, words can be counted and the importance of the words in the dataset can be determined, which can help to reduce the redundant data from the dataset. Feature Extraction helps to minimize the number of features in a dataset by generating (and then discarding the original features) new features from the current ones. Much of the details found in the original set of features should then be represented by this new reduced set of features. In this research, three types of feature extraction techniques were used namely, Count vectorizer [16], TF-IDF vectorizer [17] and Word2vector Embedding [18].

After feature extraction, the following classification methods were applied to the datasets for the detection of fake news: Support Vector Machine (SVM) [19], Logistic Regression (LR), Naïve Bayes (NB) [20], Decision Trees [21], Random Forest, K-nearest Neighbors (KNN) [22], and Recurrent Neural Networks (RNN) [23].

## 5 RESULTS AND DISCUSSION

### 5.1 Classification results for the first dataset

The classification was done on the dataset using the extracted features. For each model TF-IDF features were used for training the models by using the headline, body and combination of headline and body. To evaluate the accuracy of the classification models, 5-fold cross validation has been used by splitting the data randomly 5 times into 70% as training data and 30% as testing data. The classification accuracy of our proposed approach with that of (Agarwalla et al., 2019) is given in Table 1. The results of classification are much higher compared to the results obtained by (Agarwalla et al, 2019). In the next step, using grid search for finding the best tuning parameters for each model, the results could be improved much by using more features. The number of features, threshold for document frequency, tuning parameters for each classification model was found by using pipeline and grid search through various parameter ranges. 5-fold cross validation was used to find the optimal model. The accuracy of the model is presented in the Table 2 Performance of classifiers for the Body and Headline with AUC.

The accuracy with 5-fold cross validation on combination of headline and body for the support vector machine is 0.98. This accuracy is 15% more than the accuracy found by (Agarwalla et al, 2019). This shows that using more features and considering

**Table 1: Classification results compared with (Agarwalla et al, 2019 [4])**

| Feature set | Naive Bayes with lidstone smoothing | | Support vector machine (SVM) | | Logistic Regression | |
|---|---|---|---|---|---|---|
| | Current Study | * | Current Study | * | Current Study | * |
| Headline + body | 95.99 | 83.16 | 98.50 | 81.65 | 98.25 | 65.88 |
| Body | 96.16 | 82.53 | 98.41 | 81.65 | 98.08 | 65.88 |
| Headline | 89.22 | 68.05 | 89.31 | 66.24 | 89.22 | 66.57 |

bigrams instead of unigrams in the feature extraction will lead to much improvement in the classification model.

The 5-fold cross validation accuracy for Naive Bayes using the optimal parameters was found to be 0.963. The accuracy is much more than the one found by (Agarwalla et al, 2019). Although the maximum document frequency was found to be almost the same as what was used by them, but the difference here is that no maximum feature is considered for the number of features and bigram was used instead of unigram.

The maximum document frequency for Logistic Regression was found to be 0.75. No threshold for maximum features were considered. The bigram was found to be superior compared with unigram. The accuracy of Logistic Regression was found to be 0.986. It is much higher compared with (Agarwalla et al, 2019) which was 0.6588.

For Random Forest, the grid search results show that unigram features are superior compared with bigrams in contrast to all other classifiers which worked better with bigrams features. The optimal maximum document frequency was found to be 0.5. No maximum depth was considered for each decision tree. The random forest model with 100 estimator trees, the minimum samples for splitting each node was 2 and the maximum features in the split was taken as the square root of the number of features in the model. Gini impurity was used as criteria for fitting the Random Forest model. Gini impurity is a measure of likelihood that a new random variable being incorrectly classified. The accuracy of the Random Forest model was found to be 0.969 using the optimal tuning parameters.

For k-nearest Neighbour in the grid search, value of k = 20 which considers 20 neighbors for each observation was found to be optimal among 5 values which were tested (5, 10, 15, 20, 25). Maximum document frequency of 0.5 was selected as the threshold. The bigram was found to be superior compared with unigram. The k-nearest Neighbor classifier shows the accuracy of 0.939 with the 5-fold cross validation using optimal parameters.

The RNN-LSTM model was executed by using Adam optimizer, categorical cross entropy was used as the loss function. The model was run by keeping the batch size = 64 and was run for 10 epochs. The accuracy of the LSTM model for combination of headline + body is 0.98 % with loss equal to 0.137.

*5.1.1 Summary of results.* The optimal values found using grid search were entered for each model. The data was split randomly by 70:30, keeping 70% of the data in the model for training and setting 30% out for testing. All 5 classification models were trained, and the models were tested using test data. The performance of the models for the test data (precision, recall, F1 score and total accuracy) and confusion matrix are presented in Table 2.

Among the classifiers, support vector machines (SVM) with predicting 630 out of 643 fake news correctly and 549 out of 554 real news correctly has the highest accuracy, precision, and recall. Only 13 fake news articles were classified as real news wrongly and 5 real news were predicted as fake news wrongly. The total accuracy is 98.5%. The ROC curve for each of this classifier is presented in Figure 3 (a) ROC curve - classification results, test data (b) ROC curve for LSTM model ROC curve and in Table 2 Performance of classifiers for the Body and Headline with AUC. As we can see, the highest AUC is for SVM model with 0.9988 and LSTM with 0.9987. After that the AUC of Logistic regression is 0.9978, Random forest is 0.9977, Naïve Bayes is 0.9941 and K nearest neighbors is 0.9815. The AUC of all 6 classifiers is very close to 1.0 which shows all classifiers performed very well. Area under the curve (AUC) for the support vector machine (SVM) is the highest compared with other classifiers. The ROC curve for LSTM is drawn separately in Figure 3. The area under the curve AUC of the LSTM is almost close to SVM.

## 5.2 Classification Results for the Second Dataset

Classification analysis has been done using this big data, which include 20203 non-missing rows. Vijayaraghavan et al, (2020) have used 3-fold cross validation using this dataset. For comparison, a 3-fold cross validation was done using feature extraction methods of count vectorizer, TF-IDF vectorizer, word2vec embedding and implementing classification models of support vector machine (SVM) with C = 10-05, Logistic Regression with alpha = 0.1 and Random Forest model with number of estimators as 1000. Using these models and implementing cross validation the accuracy achieved is a bit higher (97.22%) than accuracy found by (Vijayaraghavan et al, 2020). They have claimed that the count vectorizer performed is the best in this dataset, while in our analysis TF-IDF vectorizer with (1,3) gram performs best with highest accuracy compared with other models in Table 3.

The TF-IDF was the best feature extraction method and like the analysis done by Vijayaraghavan et al, (2020) the word2vec is the worst feature extraction method for these datasets.

Grid search with pipeline was used for the second dataset also to find the optimal tuning parameters. After implementing the grid search with 5-fold cross validation for each grid, it was seen that among these parameters the maximum document frequency of 0.95 which do not set limits for document frequency, maximum feature as None which sets no limitation for maximum features, Penalty L2 with alpha equal 1e-05, normalization parameters of 12 were found to be optimal. It was seen that (1,3) gram which means including 1-gram, 2-gram and 3-gram features in the feature vector has been found as the optimal feature. The minimum document frequency of

**Table 2: Performance of classifiers for the Body and Headline with AUC**

| Classifier | headline+ body | fake observed | real observed | precision | recall | F1 | Accuracy | AUC |
|---|---|---|---|---|---|---|---|---|
| Naive Bayes | fake n = 643 | 607 | 36 | 0.98 | 0.94 | 0.96 | | |
| | real n = 554 | 12 | 542 | 0.94 | 0.98 | 0.96 | 0.96 | 0.9941 |
| SVM | fake n = 643 | 630 | 13 | 0.99 | 0.98 | 0.99 | | |
| | real n = 554 | 5 | 549 | 0.98 | 0.99 | 0.98 | 0.98 | 0.9988 |
| Logistic Regression | fake n = 643 | 626 | 17 | 0.99 | 0.97 | 0.98 | | |
| | real n = 554 | 4 | 550 | 0.97 | 0.99 | 0.98 | 0.98 | 0.9978 |
| Random Forest | fake n = 643 | 599 | 44 | 1.00 | 0.93 | 0.96 | | |
| | real n = 554 | 2 | 552 | 0.93 | 1.00 | 0.96 | 0.96 | 0.9977 |
| K-NN | fake n = 643 | 591 | 52 | 0.95 | 0.92 | 0.93 | | |
| | real n = 554 | 32 | 522 | 0.91 | 0.94 | 0.93 | 0.93 | 0.9815 |
| LSTM | fake n = 643 | 621 | 22 | 1.00 | 0.97 | 0.98 | | |
| | real n = 554 | 1 | 553 | 0.96 | 1.00 | 0.98 | 0.98 | 0.9987 |

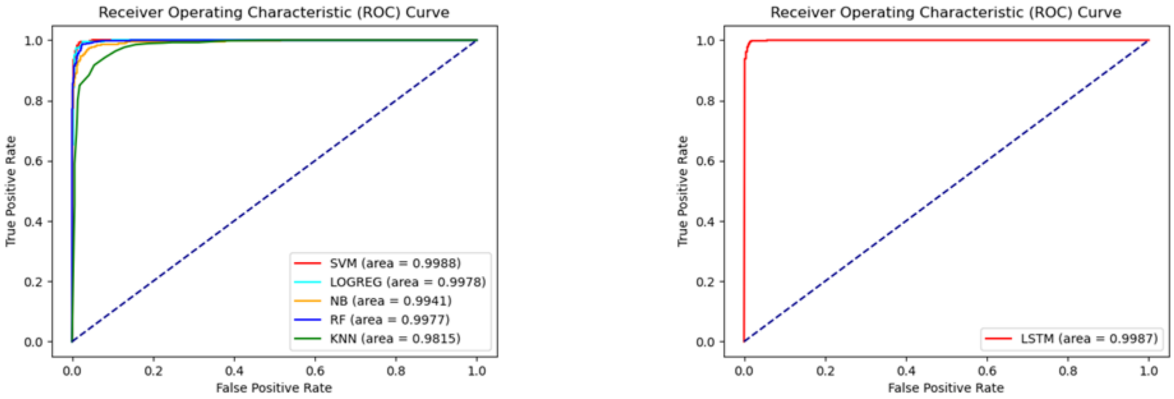


**Figure 3: : (a) ROC curve - classification results, test data (b) ROC curve for LSTM model.**

**Table 3: Results of three classification models compared with (Vijayaraghavan et al, 2020)**

| Feature set | Support vector machine | | Logistic Regression | | Random forest | |
|---|---|---|---|---|---|---|
| | Current Study | * | Current Study | * | Current Study | * |
| countvect | 95.78 | 93.06 | 97.50 | 94.45 | 96.19 | 87.64 |
| TF-IDF vect | 97.76 | 94.58 | 62.75 | 94.79 | 96.25 | 87.64 |
| word2vec | 87.11 | 91.17 | 82.00 | 91.30 | 86.3 | 88.60 |

0.007 was found as the optimal value. The accuracy of 5-fold cross validation for the support vector machine is 97.76%.

The grid search results for Naive Bayes classifier show that smoothing parameter alpha = 1.0 is optimal. The maximum document frequency was found to be 0.5. No threshold for maximum features was considered. (3,3) grams or trigram was found to be best. The 5-fold cross validation accuracy for Naive Bayes using the optimal parameters was found to be 0.90.

The optimal values for Logistic Regression were found to be L2 penalty with alpha = 1e-05. The maximum document frequency for Logistic Regression was found to be 1.0. No threshold for maximum features were considered. The (1,3) gram was found to be optimal. Results of the Count vectorizer were better compared with TF-IDF. The minimum document frequency of 0.007 was found to be a good option for removing the tokens with very low frequency. The accuracy of Logistic Regression was found to be 0.975.

In the Random Forest, values of None and 10 was used for maximum depth of tree. The maximum documents frequency of 0.1, 0.2, 0.5, 0.75 and 1.0 was tested. The maximum features of None, 100, 200, 500 and 1000 have been used. The N-Gram values of (1,1), (1,2), (1,3), (2,2) and (3,3) have been used. The optimal value of maximum features was found as 500, the (1,1) gram was found to be best in

**Table 4: Performance of classifiers for the text with AUC**

| Classifier | Body | fake observed | real observed | precision | recall | F1 | Accuracy | AUC |
|---|---|---|---|---|---|---|---|---|
| Naive Bayes | fake n = 1559 | 1542 | 17 | 0.80 | 0.99 | 0.88 | | |
| | real n = 1441 | 386 | 1055 | 0.98 | 0.73 | 0.84 | 0.87 | 0.984 |
| SVM | fake n = 1559 | 1521 | 38 | 0.98 | 0.98 | 0.98 | | |
| | real n = 1441 | 29 | 1412 | 0.97 | 0.98 | 0.98 | 0.98 | 0.995 |
| Logistic | fake n = 1559 | 1520 | 39 | 0.98 | 0.97 | 0.98 | 0.97 | |
| Regression | real n = 1441 | 36 | 1405 | 0.97 | 0.98 | 0.97 | | 0.995 |
| K-NN | fake n = 1559 | 1441 | 118 | 0.82 | 0.92 | 0.87 | | |
| | real n = 1441 | 318 | 1123 | 0.90 | 0.78 | 0.84 | 0.85 | 0.940 |
| | fake n = 1559 | 1484 | 75 | 0.99 | 0.95 | 0.97 | | |
| LSTM | real n = 1441 | 12 | 1429 | 0.95 | 0.99 | 0.97 | 0.97 | 0.996 |

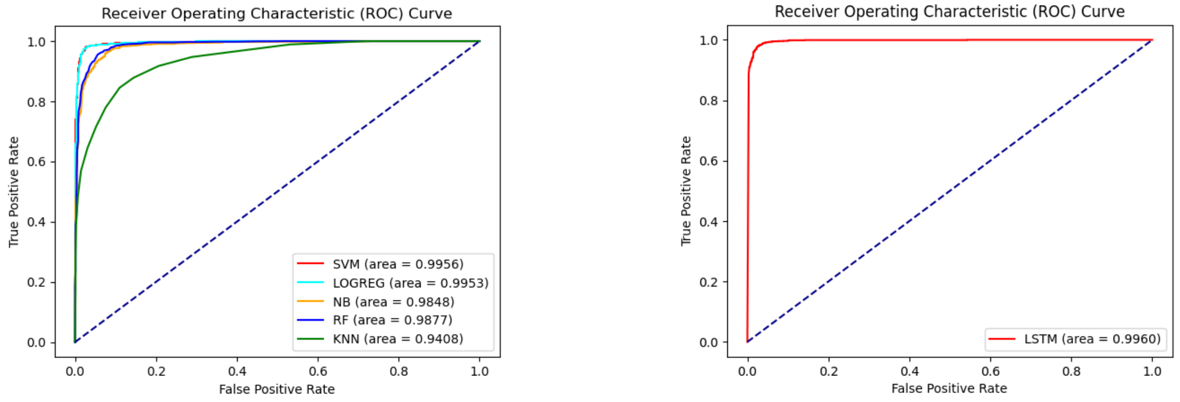


**Figure 4: (a) ROC curve, classification results-testing data (b) ROC curve for LSTM model.**

Random Forest. The maximum document frequency 1.0 was found to be optimal. No maximum depth was considered for each decision tree. The random forest model with 100 estimator trees, the minimum samples for splitting each node was 2 and the maximum features in the split was taken as the square root of the number of features in the model. Gini impurity was used as criteria for fitting the Random Forest model. The accuracy of the Random Forest model was found to be 96.19% using the optimal tuning parameters.

For k-nearest Neighbor in the grid search, k values of (5, 10, 15, 20 and 25) have been used. The options for document frequencies and N-gram were similar to Naive-Bayes. Maximum document frequency of 0.75 was selected as the threshold. The (1,3) gram was found to be superior compared with unigram and bigram. The value of K=15 was found as the optimal value. The k-nearest Neighbor classifier shows the accuracy of 84.9% with the 5-fold cross validation using optimal parameters.

The LSTM model was trained with an Adam optimizer and by using categorical cross entropy as a loss function. A batch size of 64 articles was set in each batch. The model was trained for 10 epochs. Features were generated from sequences of words, maximum number of sequences considered as 1100. The LSTM model achieved an accuracy of 97.1%, which is a bit more than the best accuracy found by Vijayaraghavan et al, (2020) using three LSTM models.

*5.2.1 Summary of Results.* The accuracy of the testing data (precision, recall, F1 score, AUC and total accuracy) and confusion matrix are presented in the Table 4 Performance of classifiers for the text with AUC. Among the classifiers, support vector machines (SVM) with predicting 1521 out of 1559 fake news correctly and 1412 out of 1441 real news correctly has the highest accuracy, precision, and recall. 38 fake news articles were classified as real news wrongly and 29 real news were predicted as fake news wrongly. The total accuracy is 97.7%. The ROC curve for each of this classifier is presented in Figure 4(a), for LSTM it is presented in Figure 4 (b) and in Table 4. As we can see, the highest AUC is for LSTM with 0.9960. The AUC of Naïve Bays is 0.9848, Logistic regression is 0.9953, Random forest is 0.9877, SVM is 0.9956 and K nearest neighbors is 0.9408.

## 6 CONCLUSIONS AND FUTURE WORK

A methodology was proposed to detect fake news using deep learning models and natural language processing (NLP). In this study, two datasets were analyzed, and classifiers were trained to develop a model that can predict the news type, whether it is fake news or real news.

The first dataset includes article news from 12 news websites. The collected data from each URL is such that all the news is fake or real from each of these 12 websites. The tagged words were used for feature extraction by count vectorizer and TF-IDF vectorizer. It was seen that TF-IDF vectorizer with (1,2) Gram (one and two consecutive tokens) gets the best features since the highest cross-validated accuracy was found by using (1,2) gram features. For classification, the Naive Bayes classifier, Support Vector Machine (SVM), Logistic Regression, K nearest Neighbours (KNN) and Random Forest were used to classify the article news as fake or real. The model with the support vector machine (SVM) achieved the highest accuracy (98.5%) in combination with body and headline and area under the ROC curve (AUC) (0.9988). Also, the LSTM model with three recurrent layers and one hidden layer was used as a deep learning method to classify the article news. The results of the LSTM model (98.03%) and ROC Curve (AUC) 0.9984. The model's accuracy considering the combination of body and headline is very high, especially in the SVM model.

Some interesting information was found using general features. It was observed that news with a greater number of words in the headline, news with more stop words used in the body part are more likely to be fake news. In contrast, news with a greater number of sentences in the body part, more words that are verbs or adjectives in the body part make the news more likely as real news. Therefore, one can conclude that real news does not tend to use big headlines, but more explanation in the body part. More sentences, especially those which include a greater number of words (as verbs or adjectives), increase the probability of being real news. More stopwords are found in the fake news compared with real news. The nouns in the body of the news articles do not increase the probability of either real or fake news. A notable thing noticed in the features was the word "photo" and "image" appeared more in the real news, while the word "video" appeared more in the fake news. Commonly used words such as "great" are more frequently seen in the fake news compared with real news. The fake news tries to pretend that it is real news, so the feature "law" can be seen more frequently in fake news than real news.

The second dataset also consisted of news articles taken from Kaggle.com, was analyzed with the same methodology, which was performed on the first dataset. For the second dataset, it was observed that nouns, adjectives, stopwords and non-alphabetic words were used more in fake news than real news. The real news had more verbs compared with fake news. In the first dataset, it was observed that the fake news has longer headlines. But in the second dataset, the results were reversed. It was seen that real news has longer headlines significantly. Therefore, this parameter differs between the two datasets. Another difference was the use of adjectives, which were significantly more in this dataset's fake news. The classification analysis was done using support vector machines, logistic regression, and random forest. The analysis used tokenizers with 1-gram to 3-gram and tuning parameters of the classifiers. The results demonstrated that 3-gram gave the best accuracy of 97.7% and ROC Curve (AUC) 0.9956 with SVM for this dataset which is Higher than results shown by Vijayaraghavan et al. (2020). Also, using LSTM 97.1% accuracy and ROC Curve (AUC) 0.9960 was achieved which is more than the accuracy achieved by Vijayaraghavan et al. (2020) which is 94.88%. For Logistic Regression 97.50% accuracy and ROC Curve (AUC) 0.9953 was achieved which is an improvement than the accuracy achieved by Vijayaraghavan et al. (2020) which is 94.79%. However, their analysis gets the count vectorizer as the best feature extraction method and word2vec as the worst performing on this dataset. This study found that word2vec with 3-gram was the best feature extraction method. It can be concluded that the use of certain features should be retained instead of removing them at the preprocessing step to get a higher accuracy of prediction, even without using deep learning models.

## 6.1 Future Work

This analysis was performed on two different datasets. Using the first dataset, some idea could be used in future studies to enhance the quality of the models:

1- Collecting data randomly from each world news website to reduce the bias in the dataset.
2- Knowing that after using the maximum document frequency for the words in the dataset, some of the words are missing, which may lead to missing some useful information for distinguishing between real and fake news. It can be considered to add a feature as several commonly used words in each news article and see its difference between real and fake news.
3- More general features could be extracted from the data like number of hashtags, number of mentions, number of each type of punctuation separately, number of characters, number of digits and unit specifiers like (kg, lbs, inches, feet and so on) to find the difference between them in real and fake news.

The second dataset looked better and was a big dataset. The analysis performed on this dataset showed some common results with the previous dataset and some reverses. Hence, it seems to be necessary to find which features are data-related and which are consistent in various data. Also, the published real and fake news behavior could be investigated by the country of release. Several independent variables can be tested at the time of release to see if the changes are time-dependent or country dependent or was just changed because the feature was not significant. It was observed that considering the tag beside the words in the tokenizing is a useful idea. This idea could be investigated more to see how good it is and how it is possible to improve the tagging portion, which is added to the token. Depending on the tag connected to the word, whether the word is noun, verb or adjective, the frequency of the word can be added for that specific tag. However, it seems important to make a relationship between these tokens, which are very similar to each other in the feature extraction phase.

## REFERENCES

[1] S. Russel and P. Norvig, Artificial Intelligence: a Modern Approach, Pearson, 2002.
[2] R. Kamble and D. Shah, "Applications of Artificial Intelligence in Human Life," International Journal of Research Granthaalayah, pp. 6(6)178-188, 2018.
[3] N. Indurkhaya and F. J. Damerau, Handbook of Natural Language Processing, Chapman & Hall, 2010.

[4] K. Agarwalla, S. Nandan, V. A. Nair and D. D. Hema, "Fake News Detection using Machine Learning and Natural Language Processing," IJRTE, pp. 7(6) 2277-3878, 2019.

[5] E. Levin, "A recurrent neural network: Limitations and training," Neural Networks, vol. 3, no. 6, pp. 641-650, 1990.

[6] X. Zhou and R. Zafarani, "A Survey of Fake News: Fundamental Theories, Detection Methods and Opportunities," ACM Computing Surveys, pp. (53) 5-40, 2021.

[7] M. S. Looijenga, "The Detection of Fake Messages using Machine," in 29th Twente Student Conference on IT, Enschede, Netherlands, 2018.

[8] UKEssays, "Analysis of Obama's Twitter and Communication Strategies in the 2012 Presidential Election," UKEssays, November 2018. [Online]. Available: https://www.ukessays.com/essays/media/analysis-of-obamas-twitter-and-communication-strategies-in-the-2012-presidential-election.php. [Accessed 16 01 2021].

[9] H. E. Wynne and Z. Z. Wint, "Content Based Fake News Detection Using N-Gram Models," in iiWAS2019: The 21st International Conference on Information Integration and Web-based Applications & Services, Munich, Germany, 2019.

[10] S. Vijayaraghavan, Y. Wang, Z. Guo, J. Voong, W. Xu, A. Nasseri, J. Cai, L. Li, K. Vuong and E. Wadhwa, "Fake News Detection with Different Models," ArXiv, no. 2003.04978, 2020.

[11] JRuvika, "Fake News Detection," kaggle, 2017. [Online]. Available: https://kaggle.com/jruvika/fake-news-detection. [Accessed 16 01 2021].

[12] "Fake News," kaggle, 2018. [Online]. Available: https://www.kaggle.com/c/fake-news. [Accessed 16 01 2021].

[13] V. Gurusamy and S. Kannan, "Preprocessing Techniques for Text Mining," in RTRICS, Theni, India, 2014.

[14] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," ArXiv, no. 0205028, 2002.

[15] T. Korenius, J. Laurikkala, K. Järvelin and M. Juhola, "Stemming and lemmatization in the clustering of finnish text documents," in CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, Washington, USA, 2004.

[16] J. Teo, "readthedocs," 2018. [Online]. Available: https://readthedocs.org/projects/socialnetwork/downloads/pdf/latest/. [Accessed 16 01 2021].

[17] A. Aizawa, "An information-theoretic perspective of tf–idf measures," Information Processing and Management, vol. 39, no. 1, pp. 45-65, 2003.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," ArXiv, no. 1310.4546 , 2013.

[19] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, no. 20, pp. 273-295, 1995.

[20] D. Berrar, "Bayes' Theorem and Naive Bayes Classifier," in Reference Module in Life Sciences, 2018, pp. 1-18.

[21] P. H. Swain and H. Hauska, "The decision tree classifier: Design and potential," IEEE Transactions on Geoscience Electronics, vol. 15, no. 3, pp. 142-147, 1977.

[22] Z. Zhang, "Introduction to machine learning: k-nearest neighbors," ATM Annals of Translational Medicine, vol. 4, no. 11, 2016.

[23] N. Kalchbrenner, I. Danihelka and A. Graves, "Grid Long Short-Term Memory," ArXiv, no. 1507.01526, 2016.

# Self-Adapting Design and Maintenance of Multi-Model Databases

Irena Holubová
Department of Software Engineering,
Charles University
Prague, Czech Republic
irena.holubova@matfyz.cuni.cz

Pavel Koupil
Department of Software Engineering,
Charles University
Prague, Czech Republic
pavel.koupil@matfyz.cuni.cz

Jiaheng Lu
Department of Computer Science,
University of Helsinki
Helsinki, Finland
jiaheng.lu@helsinki.fi

## ABSTRACT

Multi-model data is organised in various mutually interlinked formats and models, often with contradictory features. In addition, its structure may change over time, and its size can grow to the extremes of Big Data. In terms of research and practical processing, this creates one of the most complex challenges of effective data management.

As it is not humanly possible to handle such a complex task manually, in this *vision* paper, we focus on the area of automatic management of dynamic multi-model Big Data. We envision a framework capable of accepting different levels of user input and different types of data, queries, changes, and propagation strategies and ensuring the preservation of adequate and efficient data access.

## CCS CONCEPTS

• **Theory of computation** → **Data modeling**; • **Information systems** → **Data model extensions**;

## KEYWORDS

multi-model databases, evolution management, self-adapting systems, database design

**ACM Reference Format:**
Irena Holubová, Pavel Koupil, and Jiaheng Lu. 2022. Self-Adapting Design and Maintenance of Multi-Model Databases. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3548785.3548810

## 1 INTRODUCTION AND MOTIVATION

From the 388 existing database management systems (DBMSs)[1], more than 28% are classified as *multi-model*, i.e., supporting more than one logical model. If we consider only the 50 most popular representatives, involving the key players, such as, e.g., *Oracle DB*, *PostgreSQL*, *MongoDB*, Microsoft *SQL Database*, *Redis* etc., we get 60% systems with the multi-model support. This observation corresponds to the Gartner predictions [11] of supporting multiple data models in a single DBMS and it reflects the requirements of current applications.

Example 1. *An example of a multi-model scenario of an e-shop is provided in Fig. 1. The relational model (violet) contains general information about customers, whereas the graph model (blue) captures their mutual friendship. The document model (green) maintains orders bounded with particular customers using the wide-column model (red). The key/value model (yellow) bears information about customers' shopping carts. As we can see, storing each record in the best fitting model avoids impedance mismatch. A sample cross-model query over the multi-model data instance might then, e.g., be "For each customer who lives in Prague, find a friend who ordered the most expensive product among all customer's friends." [58]* □

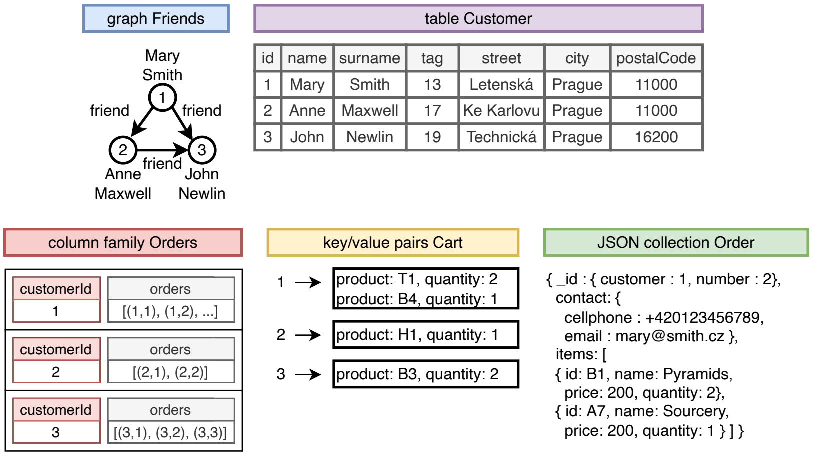
According to the extensive survey [33], the features of multi-model databases vary significantly. This status is given by the fact that (1) they are based on the distinct original core single model as well as the different target application domains, (2) there is no acknowledged standard on how to support a combination of models, cross-model querying, multi-model indices etc., and (3) the combined models have varied, often contradictory features. There are structured, semi-structured, and unstructured formats; order-preserving and order-ignorant models; systems based on strong or eventual consistency; schema-less, schema-full, and schema-mixed storage strategies; models where data normalisation is critical or where redundancy is supported etc. For all these cases, there exist multi-model representatives [13, 33, 42]. In addition, while the multi-model data by definition covers the **V**ariety feature of Big Data, many of the multi-model DBMSs are distributed and target also its high **V**olume and **V**elocity.

Example 2. *In Fig. 2 we can see an ER model of the multi-model scenario from Fig. 1. It depicts the natural first step of designing a database application to be transformed into a selected logical model. In the case of the relational model, the transformation would be more or less straightforward to avoid redundancy and null values. Or, in the case of the document model, we would also need to select the roots of the hierarchies to reflect the expected queries. However, in the case of multi-model data, the possibilities are much wider. The colours denote the combination of multiple logical models corresponding to Fig. 1.* □

The described variety of multi-model DBMSs lacking standards and the general complexity of multi-model applications indicate the main open problems of multi-model data management: (1) The initial choice of a multi-model DBMS is challenging, and the eventual later necessary migration to another system is complicated. (2) The complexity of multi-model tasks requires complex and critical decisions to be made for the optimal database design, i.e., data partitioning, (partial) schema definition, indices, (materialised) views, queries etc. (3) The naturally dynamic environment of multi-model

[1] https://db-engines.com/en/ranking

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548810



| id | name | surname | tag | street | city | postalCode |
|---|---|---|---|---|---|---|
| 1 | Mary | Smith | 13 | Letenská | Prague | 11000 |
| 2 | Anne | Maxwell | 17 | Ke Karlovu | Prague | 11000 |
| 3 | John | Newlin | 19 | Technická | Prague | 16200 |

| customerId | orders |
|---|---|
| 1 | [(1,1), (1,2), ...] |
| 2 | [(2,1), (2,2)] |
| 3 | [(3,1), (3,2), (3,3)] |

| | |
|---|---|
| 1 → | product: T1, quantity: 2<br>product: B4, quantity: 1 |
| 2 → | product: H1, quantity: 1 |
| 3 → | product: B3, quantity: 2 |

```
{ _id : { customer : 1, number : 2},
  contact: {
    cellphone : +420123456789,
    email : mary@smith.cz },
  items: [
  { id: B1, name: Pyramids,
    price: 200, quantity: 2},
  { id: A7, name: Sourcery,
    price: 200, quantity: 1 } ] }
```

**Figure 1: A multi-model scenario (inspired by [22])**

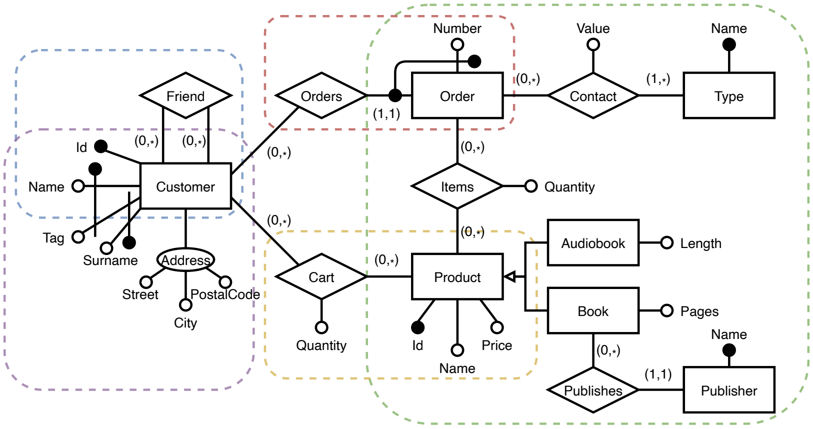


**Figure 2: An ER model of the multi-model scenario**

Big Data significantly increases the complexity of evolution management, i.e., the capability of correct and efficient adapting of a multi-model database application to the new requirements.

In this *vision* paper, to solve these problems, we envision a self-adaptive framework that would enable the design and maintenance of a multi-model database schema under the changing requirements of Big Data. We identify three levels of such a system that cover different real-world use cases and correspond to the gradual extension of the adaptability. In particular, we consider (1) user-specified changes and rule-based adaptation, (2) data-driven changes and learning-based adaptation, and, finally, (3) advanced self-adapting evolution management. We discuss the state-of-the-art as well as research challenges and tasks to be completed to reach the full robustness of the idea.

The rest of the paper is structured as follows: In Section 2, we provide an overview of related work. Section 3 discusses the three levels of adaptation the framework should gradually reach. And we conclude in Section 4.

## 2 RELATED WORK

In general, there are two approaches to manipulate and query multi-model data: (1) *polyglot persistence* and (2) *multi-model databases*. The history of polyglot persistence can be traced back to the 1970s/80s to *multi-databases* [47] and *federated databases* [14], whose main strategy is to leverage different databases to store different models of data and then develop a mediator to integrate them together in order to evaluate queries. Recently, several academic prototypes of *polystores* (e.g., *DBMS+* [31] or *BigDAWG* [10]) were also developed on the polyglot persistence paradigm. The challenge of handling the **V**ariety of Big Data has inspired a new commercially popular generation of *multi-model databases* [33], capable of storing and processing structurally different data by supporting several data models in a single DBMS having a unified query language and API. Multi-model databases manage multiple models, each being treated as a first-class citizen, with an integrated backend, which can satisfy the growing requirements for scalability, high performance, and fault tolerance.

Techniques focusing on the problem of database design can be divided into rule-based, search-based, and model-based, all of which can be general or system-specific. As summarised in [40], the first related solutions from the 1970s focused on index selection and database partitioning for relational DBMSs. Around the turn of the millennium, there was a boom of self-tuning DBMSs finding optimal database settings (i.e., indices, materialised views, partitioning schemes etc.) for a given query workload or optimal "knob" tuning (e.g., memory allocation). However, the approaches were mostly *rule-based* exploiting heuristic rules and various levels of user involvement. Unfortunately, data management heuristics can grow complex when trying to change them from specific to general cases [9]. Alternative, *search-based* approaches (e.g., [61]) search the space of possible configurations for a (sub)optimal solution.

With the dawn of Big Data entailing the introduction of novel DBMSs and the novel approaches in artificial intelligence (AI), such as the deep reinforcement learning [59] or Bayesian optimization [8], *model-based* approaches capable of learning and adapting the choice of the solution have recently appeared. However, in all the cases, the solutions are system- or data model-specific or highly limited in terms of data structures. In contrast, the area of multi-model Big Data in its full complexity is still untouched.

In general, the approaches connecting database technologies and AI can be divided into two groups [28]. *DB4AI techniques* can optimize AI models and strategies, such as, e.g., the AI-native DBMS *openGauss* [29] which supports the native AI computing engine, model management, AI operators, native AI execution plan etc. Conversely, in *AI4DB techniques*, where also the envisioned framework belongs, the AI can make DBMSs more intelligent. There are learning-based techniques exploiting, e.g., reinforcement learning or deep learning, for database *configuration*, such as an index selection [21], an index advisor [35], a partitioning advisor [17], or general knob tuning [59]; *query optimization* [50] or join order selection [36]; *design*, such as learned indices [7] or the key/value design [19]; *monitoring* predicting, e.g., query arrival rates [34] or performance [60], *security* involving, e.g., data anomaly detection [32] etc.

Recent years have also witnessed the emergence of the autonomous or self-driving DBMSs (e.g., *Oracle Autonomous Database* [39] or *Peloton* [41]) which are expected to automatically and constantly configure, tune, and optimise themselves without any intervention from human experts. Since the optimal configuration setting is highly dependent on the workload characteristics, a critical step for an autonomous DBMS is to predict the future workload based on the historical data. Firstly, the DBMS should be able to forecast when the workload will significantly change (i.e., workload shift), how many workloads will arrive (i.e., arrival rate), and what is the following query that a user will execute (i.e., next query) in the future. That predicted workload information enables an autonomous DBMS to decide when and how to re-configure itself in a predictive manner before the workload changes occur. Secondly, an autonomous DBMS also needs to predict the query performance by estimating an essential run time before execution, such as how long a query will take to complete (i.e., execution time) and how much resources will be consumed (i.e., resource utilisation). Predicting the execution time and resource demand before execution is helpful in many tasks, including query scheduling, progress monitoring, and resource management [55, 57].

In the context of related work, the envisioned idea of this paper aims to cover the area of *multi-model database maintenance in a dynamic environment*, focusing on a combination of database reconfiguration, redesign, and query rewriting utilising both rule-based and model-based approaches. In addition, we target *universally applicable multi-model generalisation* of the so far considered single-model DBMSs.

# 3 RESEARCH CHALLENGES AND ENVISIONED SOLUTIONS

The optimal database design in the context of multi-model data and its maintenance under changing requirements and conditions is a challenging aim. In the case of a simple use case, we can rely on a skilled human database administrator (DBA); however, the multi-model tasks are, in essence, complex, especially when dealing with Big Data, and thus hardly manageable manually. By extending and integrating basic research solutions, a robust framework can be designed, capable of (1) accepting different levels of user interaction as well as different types of input data, queries, and changes, (2) supporting a wide range of propagation and modification strategies reflecting specific multi-model features, and (3) ensuring the preservation of adequate and efficient data access. However, there are critical features that need to be taken into account to achieve a truly robust solution:

(1) *Universal Applicability and Portability:* The proposed approaches must be applicable to any multi-model DBMSs (or polystore), i.e., any combination of existing popular models.
(2) *Wide Range of Use Cases:* The proposed solution should cover a wide range of real-world use cases. For this purpose, it is necessary to utilise a combination of rule-based and model-based strategies based on the optional initial user settings and decisions, as well as approaches to extracting the knowledge purely from the variable input data, queries etc.
(3) *Practical Impact:* All the proposed algorithms must still preserve a tight relation to the existing systems so that they can be exploited in real-world scenarios and implementations.

In this section, we discuss three levels of the adaptation process we have identified that simultaneously form gradual extensions of the envisioned self-adapting framework.

## 3.1 Level I. User-Specified Changes and Rule-Based Adaptation

In the rest of the text, we consider the following popular data models: *aggregate-oriented* key/value, document, and wide-column, together with *aggregate-ignorant* relational, array, and graph. At the logical level, the transition between two models can be expressed either via (1) *inter-model references* or by (2) *embedding* one model into another (such as, e.g., columns of type *JSONB* in relational tables of *PostgreSQL*). Another possible combination of models is via (3) *cross-model redundancy*, i.e., storing the same data fragment in multiple models.

To "grasp" the specifics of various data models in a unified way and propose a solution that is not system-specific but can be easily transferred to another system, we need a more abstract unified representation of multi-model data. Currently, there exist several proposals, such as the NoSQL AbstractModel [3] for aggregate-oriented databases or the unified abstract representation of multi-model data based on the category theory [22, 23]. The latter enables the view of multi-model data as a categorical graph mapped to a particular combination of models. The graph can also be queried using a categorical graph query language mapped to a system-specific cross-model query language. This abstract multi-model schema can either be created manually [25] or inferred from input data [24]. Moreover, unifying categorical approaches are proven to be suitable for applications in, e.g., machine learning [46].

In this first level, we can assume that the user creates a basic multi-model system-specific database schema and transforms it to its abstract unified schema over which changes representing new requirements are then specified[2]. As the first step, we need to ensure the necessary core functionality, i.e., their correct and complete propagation to all affected parts of the system (i.e., logical data structures and data instances), including the respective operations and related structures (i.e., queries and views, indices etc.) [4]. The rule-based approach must consider the specifics of all models, all types of inter-model transitions, the transformation of data during cross-model migration etc. Moreover, it must identify cases when multiple options are possible and user input or additional information (e.g., queries, data statistics etc.) is needed. These cases are to be decided at further levels using AI approaches.

Currently, there exists several papers dealing with rule-based evolution management of single-model systems, such as XML [38, 43] or relational [6], and REST APIs [44]. There also exist approaches dealing with closely related aggregate-oriented models [16, 49]. In the case of multi-model databases, this task is more subtle and difficult since, except for intra-model changes, we have to deal with *inter-model* changes for which the single-model approaches cannot be directly re-used, together with cross-model redundancy, cross-model integrity constraints etc. Apart from the recent preliminary academic prototype *MM-evolver* [53], there are, in principle, no tools supporting schema evolution in multi-model databases in its full complexity.

To sum up, the first level of an adaptable multi-model framework needs an abstract representation of any combination of popular models. A minimal and complete set of schema modification operations (SMOs), both basic and composite, and their precise semantics must be then defined, together with an algorithm for efficient propagation to data instances. A crucial aspect of multi-model evolution management [18] is the propagation of changes to all types of inter-model transitions. The propagation strategies should also consider multi-model queries. And last but not least, a set of cost functions needs to be proposed reflecting the complexity and efficiency of each transformation regarding the given use case, i.e., data statistics, query workload, storage strategies, selected DBMS-specific features etc.

### 3.2 Level II. Data-Driven Changes and Learning-Based Adaptation

In the second level, we consider the case when the user performs the initial setting of the system and then (s)he lets the system continue (semi-)autonomously. So, in this new context, the user does not specify the changes explicitly using SMOs. They need to be extracted from the changing data and propagated to the unifying model, where all affected parts of the system can be identified. Since there is no direct user input or feedback, AI approaches need to be utilised to decide the ambiguous cases in change identification and propagation strategy.

Another source of changes that need to be utilised is the changes in queries. Such a change may need to be propagated to storage strategies to preserve/increase the efficiency of query evaluation. Besides traditional single-model strategies, such as modification of indices, the support for redundancy in distributed DBMSs, including, e.g., (cross-model) materialised views or cross-model redundancy, can be exploited.

Identifying a change from input data can be viewed as a particular case of dynamic schema inference. Even in schema-less databases, an implicit schema, i.e., a structure of the data expected by the application, exists. Hence, the idea of schemalessness is often instead characterised as *schema-on-demand* needed when the data is to be processed. However, most existing approaches assume a static input data set and focus only on single-model schema inference. A large set of schema inference approaches, both heuristic [37] and grammar-inferring [2], can be found for XML data, which is accompanied by standard schema definition languages DTD and XML Schema. There also exist papers that focus on inference of integrity constraints [54] or Schematron schemas [26]. With the dawn of NoSQL databases and the related popularity of the JSON format, there also appeared approaches inferring (Big) JSON data [1, 12] or general approaches for aggregate-oriented databases [5, 45]. The first and, to the best of our knowledge, also the only result focusing on multi-model schema-inference is the scalable academic prototype called *MM-infer* [24].

Optimising query evaluation using various AI approaches has been the focus of researchers for many years. Especially with the arrival of Big Data and deep learning approaches, a new wave of proposals has occurred. There exist proposals for, e.g., learned indices which reflect the actual distribution of data [21, 27]; join order selection based on reinforcement learning [36]; estimation of benefits of materialized views using deep reinforcement learning [15, 30]; or autonomous tuning of database knobs [52]. However, most approaches solve only the single-model case or are system-specific.

First, a multi-model schema inference approach capable of inferring a schema dynamically, i.e., for a changing data set, needs to be designed. The process must identify the changes/extensions of the data structures, including alternatives, and map them to the SMOs proposed in the first level. Second, we need to extend the propagation strategies and the respective costs with further transformations influencing query evaluation, such as modification of indices, materialised views, cross-model redundancy etc. Finally, having the set of SMOs and their alternatives, together with a set of respective propagation strategies and their costs, the framework can be extended toward autonomous decisions by exploiting supervised learning techniques to mimic the decisions of a human user in the multi-model context.

Considering the query performance, a closely related approach is represented by autonomous DBMSs expected to automatically and constantly tune themselves by adapting to data and workload

[2] If specified over the logical schema of the selected multi-model database, they can be propagated to the abstract schema.

changes. There are three major topics on workload-aware performance tuning for an autonomous DBMS: (1) workload and data classification, (2) workload and data forecasting, and (3) system tuning [56]. The goal is to enable the multi-model DBMSs to continuously and automatically adjust databases' configurations by analyzing the evolving multi-model workload and making optimal decisions for changing workload types and data. The corresponding tuning tasks consist of two main aspects: (1) multi-model database design and (2) resource provisioning. In the former one, based on the multi-model workload changes, the databases need to evolve the physical design, such as indexes, materialized views, partitioning, and storage, to achieve optimal performance. Sometimes the databases also have to re-design the multi-model schema according to the workload information. In the latter case of resource provisioning, the multi-model database must estimate the needed hardware resources to support a new workload not yet deployed in a production environment, including CPU, RAM, Disk I/O, buffer pool size, page size, etc. As we can see, the latter aspect is tightly bound with a particular DBMS and its specifics. Since we aim to provide a universal approach, another open problem is to find the balance between universal applicability and maximization of efficiency.

## 3.3 Level III. Advanced Self-Adapting Evolution Management

The previous two levels ensure that the DBMS is entirely and correctly modified to ensure the same functionality and at least the same efficiency of query evaluation within the dynamic environment. The last considered level will focus on advanced and more complex use cases (i.e., inputs) and their solutions (i.e., propagation strategies) to reach complete self-adapting robustness.

First, an important change to consider regarding the input is when there is no initial user-specified database schema, i.e., we only have the input data and queries to be efficiently evaluated. In other words, we approach the concept of a *data lake*[3]. Next, we have to consider that some of the changes detected in the data may represent errors/outliers. The occurrence of syntactically, semantically, or statistically anomalous data needs to be detected to avoid their complex but unsolicited propagation, i.e., significant unwanted changes in the system.

Second, regarding the change propagation, we have so far considered only the *eager* strategy, i.e., immediate modifications. However, as depicted in [20] for aggregate-oriented DBMSs, the *lazy* or *pro-active* strategies (i.e., based on the history of changes, they predict probable near future changing parts) may bring significant benefits for selected use cases. So, the search space of options is much more complex.

Currently, there exist approaches that directly learn database schema design (for single-model systems) – e.g., the one designing the relational schema using the deep reinforcement learning [17]. Or, the *data structure alchemy* [19] defines the design space for a key/value store using database knob tuning. Considering the multi-model related work again, probably the first and only related solution that (lazily, eagerly, or pro-actively) reflects the changes in user requirements has been proposed and evaluated for aggregate-oriented DBMSs within systems *Darwin* and *MigCast* [20]. Another academic project *OctopusDB*[4] can automatically evolve its storage and execution architecture over time based on the application's workload. However, it does not target multiple models but multiple systems, namely OLTP, OLAP, streaming systems, and scan-oriented DBMSs.

The problem of detection of errors is solved in many areas, such as, e.g., *data curation* [48] and *data discovery* [51], i.e., the process of discovering data that are relevant for specific tasks, where the data first need to be cleaned. However, most data-curation solutions cannot be easily fully automated, as they are often ad-hoc and require substantial human effort. Hence, using AI techniques learning from the history of a dynamic system seems to be a natural approach and extension of the proposed framework.

The described broadening of both input and respective propagation of changes will enable reaching the solution's target robustness. First, based on multi-model schema inference approaches, a strategy inferring the logical multi-model database schema should be proposed. It will be designed concerning the given query workload and partially utilising the costs of propagation strategies. The propagation strategies and their cost evaluation can be extended with advanced multi-model features, such as modification of multi-model indices, inter-model data migration, cross-model materialised views, cross-model redundancy etc. In addition, the eager propagation of changes can be extended with lazy and pro-active options, as inspired by data migration systems *Darwin* and *MigCast* [20] or the autonomous self-driving DBMSs [55, 57] and the appropriateness of their application experimentally evaluated. The approach for detecting changes needs to be extended to detect errors/outliers in the data. AI techniques such as anomaly detection can be utilised for this purpose. Finally, based on the formal model of the system, possible changes, and their cost, a fully automated approach to updating the database schema can be proposed by exploiting automated planning techniques.

## 3.4 Summary

To sum up the ideas, in Fig. 3 we depict the process of gradual extension of the framework, i.e., its three levels. At the same time, it represents several use cases that we cover. As we can see, user involvement (red) in the design and maintenance of the multi-mode database schema (green) decreases with the growing level. The initial setting and the SMOs are gradually replaced by providing only input multi-model data to be managed and respective cross-model queries. The framework (yellow) becomes more sophisticated (depicted using the blue color) as it adopts AI approaches to mimic the user's decisions. It can also make more complex decisions in terms of processing of the input (e.g., detection of outliers), supported transformations (e.g., cross-model data migration), or propagation strategies (i.e., lazy or pro-active) that exceed the abilities of a human DBA, especially considering the world of multi-model Big Data.

[3] https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/

[4] https://dbdb.io/db/octopusdb

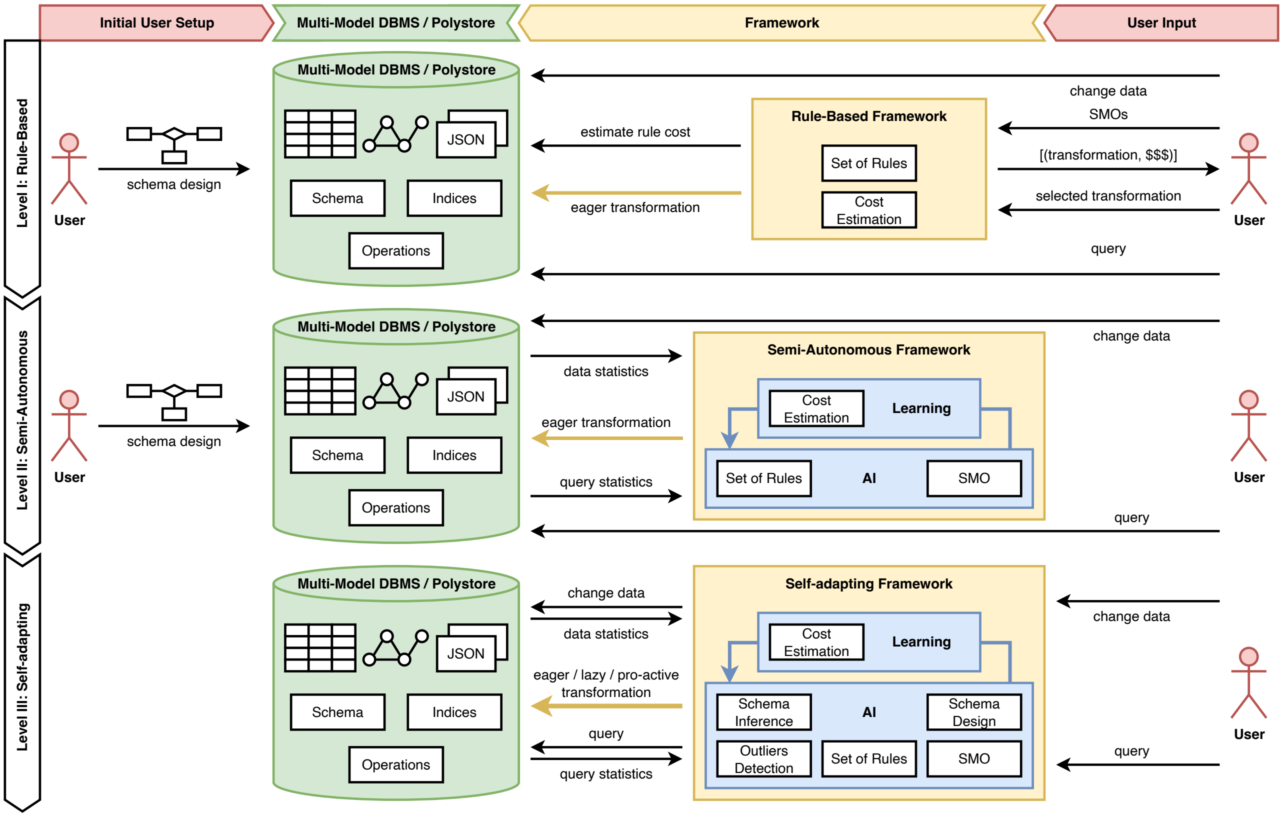


Figure 3: Levels and use cases of the self-adapting framework

## 4 CONCLUSION

This paper aimed to envision a self-adapting evolution management framework that can eventually maintain and design an optimal multi-model database schema and related structures. Since such a framework is a complex tool, we propose three levels that correspond to three steps, gradually extending a necessary basic functionality towards the robust target. In other words, we consider various real-world use cases. First, the user-defined changes and rule-based adaptation ensure the core functionality. It corresponds to the initial situation when the particular use case is not that complex and still manageable by a skilled DBA. Next, we assume the case when the user input is no longer available. The changes need to be extracted from the changing data, and the choice of the optimal propagation strategy needs to be decided using AI strategies. Finally, we consider the case when the user input is not available from the beginning, and there can be errors/outliers in the input data to be detected. Also, the propagation of changes can be smart, i.e., lazy or proactive, aiming to minimise the impact of the changes.

The proposed levels and their parts can be solved separately, reflecting the needs of a subset of use cases. Also, in many aspects, several single-model or system-specific approaches exist that can serve as a verified basis. However, the multi-model environment requires consideration of more complex situations and their more complex solutions.

## ACKNOWLEDGMENTS

The work was supported by the GAČR project no. 20-22276S and by the Charles University project GA UK no. 16222.

## REFERENCES

[1] Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, and Carlo Sartiani. 2019. Parametric Schema Inference for Massive JSON Datasets. *The VLDB Journal* 28, 4 (2019). https://doi.org/10.1007/s00778-018-0532-7

[2] Geert Jan Bex, Wouter Gelade, Frank Neven, and Stijn Vansummeren. 2010. Learning Deterministic Regular Expressions for the Inference of Schemas from XML Data. *ACM Trans. Web* 4, 4, Article 14 (Sept. 2010), 32 pages. https://doi.org/10.1145/1841909.1841911

[3] Francesca Bugiotti, Luca Cabibbo, Paolo Atzeni, and Riccardo Torlone. 2014. Database Design for NoSQL Systems. In *Conceptual Modeling*. Springer, Cham, 223–231.

[4] Loredana Caruccio, Giuseppe Polese, and Genoveffa Tortora. 2016. Synchronization of Queries and Views Upon Schema Evolutions: A Survey. *ACM Trans. Database Syst.* 41, 2, Article 9 (may 2016), 41 pages. https://doi.org/10.1145/2903726

[5] Alberto Hernández Chillón, Severino Feliciano Morales, Diego Sevilla, and Jesús García Molina. 2017. Exploring the Visualization of Schemas for Aggregate-Oriented NoSQL Databases. In *ER Forum/Demos 1979*. CEUR-WS.org, 72–85. http://ceur-ws.org/Vol-1979/paper-11.pdf

[6] Martin Chytil, Marek Polák, Martin Nečaský, and Irena Holubová. 2013. Evolution of a Relational Schema and Its Impact on SQL Queries. In *IDC 2013*. Springer, 5–15. https://doi.org/10.1007/978-3-319-01571-2_2

[7] Jialin Ding, Umar Farooq Minhas, Jia Yu, Chi Wang, et al. 2020. ALEX: An Updatable Adaptive Learned Index. In *SIGMOD 2020*. ACM, 969–984. https://doi.org/10.1145/3318464.3389711

[8] Songyun Duan, Vamsidhar Thummala, and Shivnath Babu. 2009. Tuning Database Configuration Parameters with iTuned. *Proc. VLDB Endow.* 2, 1 (2009), 1246–1257. https://doi.org/10.14778/1687627.1687767

[9] Gabriel Campero Durand. 2019. *AI Techniques for Database Management (AI4DB).* Otto-von-Guericke University of Magdebur. https://www.dbse.ovgu.de/en/-p-578-EGOTEC-jjlju9r889k5nsvcuqqa6667f0/_/5_ai-1.pdf.

[10] Aaron J. Elmore, Jennie Duggan, Mike Stonebraker, Magdalena Balazinska, et al. 2015. A Demonstration of the BigDAWG Polystore System. *PVLDB* 8, 12 (2015), 1908–1911.

[11] Donald Feinberg, Merv Adrian, Nick Heudecker, Adam M. Ronthal, et al. 12 October 2015. Gartner Magic Quadrant for Operational Database Management Systems, 12 October 2015.

[12] Enrico Gallinucci, Matteo Golfarelli, and Stefano Rizzi. 2018. Schema Profiling of Document-Oriented Databases. *Inf. Syst.* 75 (2018), 13–25. https://doi.org/10.1016/j.is.2018.02.007

[13] Daniel Glake, Felix Kiehn, Mareike Schmidt, Fabian Panse, and Norbert Ritter. 2022. Towards Polyglot Data Stores – Overview and Open Research Questions. *arXiv preprint arXiv:2204.05779* (2022).

[14] Michael Hammer and Dennis McLeod. 1979. *On Database Management System Architecture.* MIT, Laboratory for Computer Science.

[15] Yue Han, Guoliang Li, Haitao Yuan, and Ji Sun. 2021. An Autonomous Materialized View Management System with Deep Reinforcement Learning. In *ICDE 2021.* IEEE, 2159–2164. https://doi.org/10.1109/ICDE51399.2021.00217

[16] Andrea Hillenbrand, Uta Störl, Maksym Levchenko, Shamil Nabiyev, et al. 2020. Towards Self-Adapting Data Migration in the Context of Schema Evolution in NoSQL Databases. In *ICDE Workshops 2020.* IEEE, 133–138. https://doi.org/10.1109/ICDEW49219.2020.00013

[17] Benjamin Hilprecht, Carsten Binnig, and Uwe Röhm. 2020. Learning a Partitioning Advisor for Cloud Databases. In *SIGMOD 2020.* ACM, 143–157. https://doi.org/10.1145/3318464.3389704

[18] Irena Holubová, Michal Vavrek, and Stefanie Scherzinger. 2021. Evolution Management in Multi-Model Databases. *Data Knowl. Eng.* 136, 101932 (2021). https://doi.org/10.1016/j.datak.2021.101932

[19] Stratos Idreos, Niv Dayan, Wilson Qin, Mali Akmanalp, et al. 2019. Design Continuums and the Path Toward Self-Designing Key-Value Stores that Know and Learn. In *CIDR 2019.* www.cidrdb.org. http://cidrdb.org/cidr2019/papers/p143-idreos-cidr19.pdf

[20] Meike Klettke, Uta Störl, Manuel Shenavai, and Stefanie Scherzinger. 2016. NoSQL Schema Evolution and Big Data Migration at Scale. In *BigData 2016.* IEEE, 2764–2774.

[21] Jan Kossmann, Stefan Halfpap, Marcel Jankrift, and Rainer Schlosser. 2020. Magic Mirror in My Hand, Which Is the Best in the Land? An Experimental Evaluation of Index Selection Algorithms. *Proc. VLDB Endow.* 13, 11 (2020), 2382–2395. http://www.vldb.org/pvldb/vol13/p2382-kossmann.pdf

[22] Pavel Koupil and Irena Holubová. 2022. A Unified Representation and Transformation of Multi-Model Data using Category Theory. *J. of Big Data (accepted)* (2022).

[23] Pavel Koupil and Irena Holubová. 2022. Unifying Categorical Representation of Multi-Model Data. In *SAC 2022.* ACM, 365–371.

[24] Pavel Koupil, Sebastian Hricko, and Irena Holubová. 2022. MM-infer: A Tool for Inference of Multi-Model Schemas. In *EDBT 2022.* OpenProceedings.org. https://www.ksi.mff.cuni.cz/~koupil/mm-infer/index.html

[25] Pavel Koupil, Martin Svoboda, and Irena Holubová. 2021. MM-cat: A Tool for Modeling and Transformation of Multi-Model Data using Category Theory. In *MODELS 2021.* IEEE, 635–639. https://www.ksi.mff.cuni.cz/~koupil/mm-cat/index.html

[26] Michal Kozák, Jakub Stárka, and Irena Mlýnková. 2012. Schematron Schema Inference. In *IDEAS 2012.* ACM, 42–50. https://doi.org/10.1145/2351476.2351482

[27] Tim Kraska, Alex Beutel, Ed H Chi, Jeffrey Dean, et al. 2018. The case for learned index structures. In *SIGMOD 2018.* 489–504.

[28] Guoliang Li, Xuanhe Zhou, and Lei Cao. 2021. AI Meets Database: AI4DB and DB4AI. In *SIGMOD 2021.* ACM, 2859–2866. https://doi.org/10.1145/3448016.3457542

[29] Guoliang Li, Xuanhe Zhou, Ji Sun, Xiang Yu, et al. 2021. openGauss: An Autonomous Database System. *Proc. VLDB Endow.* 14, 12 (2021), 3028–3041. http://www.vldb.org/pvldb/vol14/p3028-li.pdf

[30] Xi Liang, Aaron J. Elmore, and Sanjay Krishnan. 2019. Opportunistic View Materialization with Deep Reinforcement Learning. *CoRR* abs/1903.01363 (2019). arXiv:1903.01363 http://arxiv.org/abs/1903.01363

[31] Harold Lim, Yuzhang Han, and Shivnath Babu. 2013. How to Fit when No One Size Fits. In *CIDR 2013.* www.cidrdb.org.

[32] Zhechao Lin, Xiang Li, and Xiaohui Kuang. 2017. Machine Learning in Vulnerability Databases. In *ISCID 2017, Volume 1.* IEEE, 108–113. https://doi.org/10.1109/ISCID.2017.24

[33] Jiaheng Lu and Irena Holubová. 2019. Multi-model Databases: A New Journey to Handle the Variety of Data. *ACM Comput. Surv.* 52, 3 (2019), 38 pages.

[34] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, et al. 2018. Query-based Workload Forecasting for Self-Driving Database Management Systems. In *SIGMOD 2018.* ACM, 631–645. https://doi.org/10.1145/3183713.3196908

[35] Lin Ma, Bailu Ding, Sudipto Das, and Adith Swaminathan. 2020. Active Learning for ML Enhanced Database Systems. In *SIGMOD 2020.* ACM, 175–191. https://doi.org/10.1145/3318464.3389768

[36] Ryan Marcus and Olga Papaemmanouil. 2018. Deep Reinforcement Learning for Join Order Enumeration. In *aiDM@SIGMOD 2018.* ACM, 3:1–3:4. https://doi.org/10.1145/3211954.3211957

[37] Irena Mlýnková and Martin Nečaský. 2013. Heuristic Methods for Inference of XML Schemas: Lessons Learned and Open Issues. *Informatica, Lith. Acad. Sci.* 24, 4 (2013), 577–602. http://content.iospress.com/articles/informatica/inf24-4-05

[38] M. Nečaský, I. Mlýnková, J. Klímek, and J. Malý. 2012. When Conceptual Model Meets Grammar: A Dual Approach to XML Data Modeling. *Data & Knowledge Engineering* 72 (2012), 1–30.

[39] Oracle. 2022. Oracle Autonomous Database. https://www.oracle.com/autonomous-database/.

[40] Andy Pavlo. 2018. *What is a Self-Driving Database Management System?* A. Pavlo blog. https://www.cs.cmu.edu/~pavlo/blog/2018/04/what-is-a-self-driving-database-management-system.html.

[41] Andrew Pavlo, Gustavo Angulo, Joy Arulraj, Haibin Lin, et al. 2017. Self-Driving Database Management Systems. In *CIDR 2017.* www.cidrdb.org.

[42] Ewa Pluciennik and Kamil Zgorzalek. 2017. The Multi-model Databases – A Review. In *BDAS 2017 (Communications in Computer and Information Science, Vol. 716).* 141–152.

[43] Marek Polák, Martin Chytil, Karel Jakubec, Vladimir Kudelas, et al. 2015. Data and Query Adaptation Using DaemonX. *Computing and Informatics* 34, 1 (2015), 99–137. http://www.cai.sk/ojs/index.php/cai/article/view/2040

[44] Marek Polák and Irena Holubová. 2015. REST API Management and Evolution Using MDA. In *C3S2E 2015.* ACM, 102–109. https://doi.org/10.1145/2790798.2790820

[45] Diego Sevilla Ruiz, Severino Feliciano Morales, and Jesús García Molina. 2015. Inferring Versioned Schemas from NoSQL Databases and Its Applications. In *ER 2015.* Springer, Cham, 467–480.

[46] Dan Shiebler, Bruno Gavranović, and Paul Wilson. 2021. Category Theory in Machine Learning. *arXiv:2106.07032* (2021).

[47] John Miles Smith, Philip A. Bernstein, Umeshwar Dayal, Nathan Goodman, et al. 1981. Multibase: Integrating Heterogeneous Distributed Database Systems. In *AFIPS 1981* (Chicago, Illinois). ACM, New York, NY, USA, 487–499. https://doi.org/10.1145/1500412.1500483

[48] Michael Stonebraker, Daniel Bruckner, Ihab F Ilyas, George Beskales, et al. 2013. Data Curation at Scale: The Data Tamer System.. In *CIDR 2013*, Vol. 2013. www.cidrdb.org.

[49] Pablo Suárez-Otero, Michael J. Mior, María José Suárez Cabal, and Javier Tuya. 2020. Maintaining NoSQL Database Quality During Conceptual Model Evolution. In *BigData 2020.* IEEE, 2043–2048. https://doi.org/10.1109/BigData50022.2020.9378228

[50] Ji Sun and Guoliang Li. 2019. An End-to-End Learning-based Cost Estimator. *Proc. VLDB Endow.* 13, 3 (2019), 307–319. https://doi.org/10.14778/3368289.3368296

[51] Saravanan Thirumuruganathan, Nan Tang, Mourad Ouzzani, and AnHai Doan. 2020. Data Curation with Deep Learning.. In *EDBT 2020.* OpenProceedings.org, 277–286.

[52] Dana Van Aken, Andrew Pavlo, Geoffrey J Gordon, and Bohan Zhang. 2017. Automatic Database Management System Tuning through Large-Scale Machine Learning. In *SIGMOD 2017.* ACM, 1009–1024.

[53] Michal Vavrek, Irena Holubová, and Stefanie Scherzinger. 2019. MM-evolver: A Multi-model Evolution Management Tool. In *EDBT 2019.* OpenProceedings.org, 586–589. https://doi.org/10.5441/002/edbt.2019.62

[54] Matej Vitásek and Irena Mlýnková. 2012. Inference of XML Integrity Constraints. In *ADBIS 2012.* Springer, 285–296. https://doi.org/10.1007/978-3-642-32741-4_26

[55] Wentao Wu, Yun Chi, Shenghuo Zhu, Jun'ichi Tatemura, Hakan Hacigümüs, and Jeffrey F. Naughton. 2013. Predicting Query Execution Time: Are Optimizer Cost Models Really Unusable?. In *ICDE 2013.* IEEE Computer Society, 1081–1092.

[56] Zhengtong Yan, Jiaheng Lu, Naresh Chainani, and Chunbin Lin. 2021. Workload-Aware Performance Tuning for Autonomous DBMSs. In *ICDE 2021.* IEEE, 2365–2368.

[57] Zhengtong Yan, Jiaheng Lu, Qingsong Guo, Gongsheng Yuan, Calvin Sun, and Steven Yang. 2022. Make Wise Decisions for Your DBMSs: Workload Forecasting and Performance Prediction Before Execution. In *DASFAA-2022 (accepted).* Springer.

[58] Chao Zhang, Jiaheng Lu, Pengfei Xu, and Yuxing Chen. 2018. UniBench: A Benchmark for Multi-Model Database Management Systems. In *TPCTC.*

[59] Ji Zhang, Ke Zhou, Guoliang Li, Yu Liu, et al. 2021. CDBTune+: An Efficient Deep Reinforcement Learning-Based Automatic Cloud Database Tuning System. *VLDB J.* 30, 6 (2021), 959–987. https://doi.org/10.1007/s00778-021-00670-9

[60] Xuanhe Zhou, Ji Sun, Guoliang Li, and Jianhua Feng. 2020. Query Performance Prediction for Concurrent Queries using Graph Embedding. *Proc. VLDB Endow.* 13, 9 (2020), 1416–1428. https://doi.org/10.14778/3397230.3397238

[61] Yuqing Zhu, Jianxun Liu, Mengying Guo, Yungang Bao, et al. 2017. BestConfig: Tapping the Performance Potential of Systems via Automatic Configuration Tuning. In *SoCC 2017.* ACM, 338–350. https://doi.org/10.1145/3127479.3128605

# A comparative study of three deep learning models for $PM_{2.5}$ interpolation

Lixin Li
Computer Science Department,
Georgia Souther University
Statesboro, GA, USA
lli@georgiasouthern.edu

Weitian Tong
Computer Science Department,
Georgia Souther University
Statesboro, GA, USA
wtong.research@gmail.com

Adolphe Some
Computer Science Department,
Georgia Souther University
Statesboro, GA, USA
as25175@georgiasouthern.edu

## ABSTRACT

$PM_{2.5}$ is a pollutant particulate matter with diameter less than 2.5 micrometer. There exist many stations installed in the world to measure its concentration. Some areas without any proper equipment nor any station installation must rely on interpolation techniques to approximate its concentration. So, there is a need of interpolation technique to approximate the concentration of the pollutant in those areas. The faster and more accurate interpolation technique can help identify more polluted areas and thus efficiently take some measures to reduce $PM_{2.5}$ harmful effects. We explored three different neural networks, i.e., Bidirectional-Long Short-Term Memory (Bi-LSTM), Gated Recurrent Unit (GRU), Temporal Convolutional Neural Networks (TCN), to interpolate the $PM_{2.5}$ concentration over the southeast region of the U.S. We investigate different data preprocessing techniques and the effects of spatiotemporal correlation on the models. We finally compare these models and make a choice on the model that is more appropriate for $PM_{2.5}$ interpolation.

## CCS CONCEPTS

• **Applied computing → Environmental sciences**; • **Computing methodologies → Supervised learning by regression**; **Neural networks**.

## KEYWORDS

Fine particle matter $PM_{2.5}$, Interpolation, Bidirectional-Long Short-Term Memory, Gated Recurrent Unit, Temporal Convolutional Neural Networks

**ACM Reference Format:**
Lixin Li, Weitian Tong, and Adolphe Some. 2022. A comparative study of three deep learning models for $PM_{2.5}$ interpolation. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3548785.3548809

## 1 INTRODUCTION

$PM_{2.5}$ are particulate matters with a diameter smaller than 2.5 $\mu$m. The indoor $PM_{2.5}$ particles originate from indoor burning activities such as smoking, cooking, operating fireplaces, and fuel-burning heaters [24]. The outdoor $PM_{2.5}$ particles are the most common. They originate from all kinds of burning activities in the environment such as vehicles and machines with fuel-based engines, industries, wildfires, and even volcanic eruptions [6]. Most particles are formed in the atmosphere as a result of complex reactions of chemicals such as sulfur dioxide and nitrogen oxides, which are pollutants emitted from power plants, industries, and automobiles [6]. They have been studied and proven to be responsible for many cardiovascular problems such as irregular heartbeat, aggravated asthma, decreased lung function, increase respiratory symptoms as irritation of the airways, coughing or difficulty breathing [6, 24, 26].

To monitor and reduce the $PM_{2.5}$ pollution, many countries have installed monitoring stations. The obtained data can be later analyzed and lead to make decisions against $PM_{2.5}$ pollution. The United States Environmental Protection Agency (EPA) established Nation Ambient Air Quality Standards for $PM_{2.5}$ since 1990 [1]. Unfortunately, many polluted places are not equipped with monitoring stations. Sometimes, places with monitoring stations don't record any data for months due to outage or technical issues. To estimate the missing or non-recorded data at some spatial locations for a specific time, effective interpolation techniques are needed.

Spatial interpolations can be grouped in two types, i.e., point interpolation (based on field data) and area interpolation (based on entity data) [18]. Point interpolations can be further divided into two sub-parts which are the exact point interpolation and approximate point interpolation. The most popular exact point interpolation techniques are Inverse Distance Weighting, kriging and shape functions [11, 12]. Spatiotemporal methods incorporate time and space simultaneously by using the known spatiotemporal measurements in the interpolation algorithm [13, 14, 21].

Qiao et al. [19] applied a hybrid of wavelet transform, stacked autoencoder and LSTM to predict $PM_{2.5}$. Bamane et al. [3] used linear interpolation, spearman's rank-order correlation, LSTM and stacked LSTM, Bi-LSTM to estimate $PM_{2.5}$ concentration. Huang et al. [7] proposed an ensemble of Convolutional Neural Network (CNN) and LSTM to estimate the concentration of $PM_{2.5}$. These models fails to capture the spatiotemporal correlation in the dataset. Tong et al. [25] proposed a spatiotemporal technique that takes in account the relationship between each monitoring site and its k-nearest neighbors and also between each monitoring site and its own measurements at different days.

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548809

We conduct a comparative study of three different neural networks, i.e., Bidirectional-Long Short-Term Memory (Bi-LSTM), Gated Recurrent Unit (GRU), Temporal Convolutional Neural Networks (TCN), to estimate the concentration of $PM_{2.5}$ over the southeast region of the U.S. We explore multiple data preprocessing techniques and the effects of spatiotemporal correlation on the models.

In the sequel, Section 2 introduces Bi-LSTM, GRU, and TCN in detail. Section 3 describes the dataset and the preprocessing techniques. Section 4 shows our experimental framework and setup. Section 5 gives the results of our experiments. We conclude in Section 6.

## 2 METHODS

A Recurrent neural networks (RNN) is a development of neural network with a feedback (recurrence) to itself. While non-recurrent unit can not receive inputs from it previous state (timestep), a recurrent unit receives inputs from its previous timestep and outputs a feedback data for its next timestep. RNN can encode dependencies between inputs but has a problem when handling long data sequences. When encoding long data dependencies, the back-propagation of the signal goes through multiples layers of neural networks. As the signal travels through more layers using certain $activation\ functions$, the gradients of the $loss\ function$ increases exponentially or vanishes, making the neural network unable to learn. This is called the vanishing gradient problem. The simplest solutions are to use the right $activation\ functions$ or perform a $batch\ normalization$. A $batch\ normalization$ is a technique to standardize the data. It is believed to make the neural network stable and faster during the training.

*Long short-term memory recurrent neural networks* (LSTM) networks are built to overcome problems associated with the long-term problems associated with the RNN [23]. It comprises of different gates. There is an input gate for the input layer, a forget gate, and an output gate [16]. The cell state and the gates are the core concept of LSTM. This is because the cell state allows for the transportation of relative information to the sequence chain. It serves as the memory of the network. The cell state makes it possible to store and transport relevant information throughout the processing of the sequence. Necessitating the availability of information from past steps and ensuring they get to later steps. Hence reducing the effects of short-term memory [7]. The cell state gets more information throughout its journey. Some information is also removed via the gates. The gates decide what to retain and what to forget in this journey.

*Gated Recurrent Unit* (GRU) is a recurrent neural network similar to LSTM but lacks an output gate and has fewer parameters which aims to solve the vanishing gradient problem experienced in LSTM [16]. GRU only have hidden states. It also only has two gates, a reset gate, and an update gate. The gates are regulating the flow of information flowing through and that allow the GRU to solve the vanishing gradient problem of a standard RNN [4]. The update and reset gate are vectors deciding the information that goes through to the output [7]. Those gates store and filter the information. The update gate in the model is used to determine the degree of the past information from past steps to carry to the future. This means that it can copy relevant information from the past and get rid of the risk of the vanishing gradient. The reset gate decides how much of the past data is irrelevant and worth forgetting [20]. Both the update and reset gates use the same formula. The only difference is the weights and gates usage. The gates affect the final output of the model. The last step in the unit consists of a vector that transfers information to the network from the current unit. To transfer information, the update gate is needed because it is responsible for determining what to collect from the current memory content [7]. GRU eradicates the vanishing gradient problem because it basically does not wash out the new input every single time but rather stores the relevant information and passes it down to future-forward steps of the network. Having also fewer operations allows GRU to be faster to train compared to LSTM.

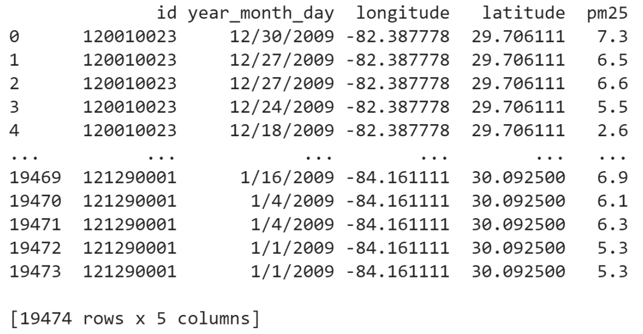
*Bidirectional recurrent neural network* (Bi-LSTM) occurs when Long Short-Term Memory (LSTM) and Bi-directional Recurrent Networks (Bi-RNN) are combined. This structure makes it possible for networks to access backward and forward information on sequences at every level. The Bi-RNN can handle inputs information from both the back and the front. Bidirectional allows the two inputs to operate one from the future to the past and another from the past to the future. Recent years have seen an increase in approaches combining recommendation systems and deep learning [7]. Merging the LSTM and Bi-RNN increases the storage capability in LSTM cell memory and the access information abilities of Bi-RNN hence making the Bi-LSTM better [23]. Bi-LSTM ability to handle data with long-range dependence allows them to improve performance on sequence classification problems [4].

*Convolutional Neural networks* (CNN) is similar to a feedforward neural network with the difference that a CNN has one or more convolutional layers. [2] A convolutional layer is similar to a hidden layer of an FNN but it uses one or more filters. Filters have input weights and generate an output neuron. The goal of the convolutional layer is to extract features mostly from input image and preserve the spatial relationship by learning features using small squares of input data [17]. CNN were first employed to learn the correlation between image and sentence [22]. However, a variation of $CNN$ called $Temporal\ Convolution\ Networks$ (TCN) has been developed for sequence modelling tasks.

*Temporal Convolutional Neural Networks* (TCN) consist of dilated, causal $1D$ (one dimensional) convolutional layers used to convulse the output with the past elements of a sequence. This allows TCN to be effective in sequence predictions hence their utilization in weather predictions [15]. A $1D$ convolutional network takes as input a 3-dimensional tensor and also outputs a 3-dimensional tensor. One single $1D$ convolutional layer receives a unique input tensor and outputs a tensor of similar unique traits to ensure an output tensor has the same length as the input tensor, zero paddings could be applied [9]. In a forecasting model, the value of a specific entry in the output depends on all previous entries in the input. This is made possible when the receptive field has the same size input length. Most convolutional hidden layers end with a pooling layer whose job it is to distill the output of the last convolutional layer to the most important elements. Temporal convolutional networks do not have time steps. They treat the temporal data as a sequence over which convolutional read operations can be performed. TCN are combined with RNN in the segmentation of video-based action by filtering low-level features which are responsible for encoding spatial-temporal information and segregating features into a classifier capturing high-level temporal information using RNN. At least two convolutional layers are needed for video-based segmentation [10]. The TCN approach for catching two levels of information hierarchically is known as the encoder-decoder framework [8].

## 3 EXPERIMENTAL DATASET

The dataset is a daily measurement of PM$_{2.5}$ at Florida in 2009. There are 19,475 rows of data, which were collected from 53 U.S. EPA's Air Quality System (AQS) monitoring sites. Each row consists of 5 columns, i.e., site id, timestamp, latitude, longitude and daily concentration PM$_{2.5}$ measurement. Sample raw data is shown in Figure 1. The time range for this dataset is between January 1st 2009 and December 31st 2009.



```
              id year_month_day  longitude   latitude  pm25
0      120010023     12/30/2009 -82.387778  29.706111   7.3
1      120010023     12/27/2009 -82.387778  29.706111   6.5
2      120010023     12/27/2009 -82.387778  29.706111   6.6
3      120010023     12/24/2009 -82.387778  29.706111   5.5
4      120010023     12/18/2009 -82.387778  29.706111   2.6
...          ...            ...        ...        ...   ...
19469  121290001      1/16/2009 -84.161111  30.092500   6.9
19470  121290001       1/4/2009 -84.161111  30.092500   6.1
19471  121290001       1/4/2009 -84.161111  30.092500   6.3
19472  121290001       1/1/2009 -84.161111  30.092500   5.3
19473  121290001       1/1/2009 -84.161111  30.092500   5.3

[19474 rows x 5 columns]
```

**Figure 1: Original Data**

We observe that some sites have duplicates on some daily records and some sites don't have some daily records. The dataset is preprocessed for the models to better capture the spatial and temporal relations.

We first drop the duplicates, then group the dataset by id and keep the first 32 *ids* with the most daily measurements. After that, we calculate the mean of PM$_{2.5}$ concentrations at each site. We then group the data by the site ids and iterated to compare the date in each group to the 365 *days* of the year. If a certain day is not present in the group of same *id*, we create a row with the missing day and complete the rest of the columns with the mean of PM$_{2.5}$. Such function is used to fill the missing days for all of our dataset. We assume the PM$_{2.5}$ concentrations at one site is related to the PM$_{2.5}$ concentrations at neighboring sites. Thus, a k-d tree-based $k$-nearest neighbor algorithm is applied to find $k$ nearest neighbors for each site and then we reshape the dataset by treating these neighbors' features as new features for the current site. In our experiments, we assume $k \in \{1, 2, 3, 4, 5, 6\}$. Once the preprocessing is complete, we have 6 datasets, one for each $k$.

## 4 EXPERIMENTS

Four performance measures are used. They are the mean absolute error *MAE*, the root-mean-square error RMSE, the mean absolute percentage error *MAPE* and the mean-square error MSE, whose computations are as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|O_i - P_i|$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_i - P_i)^2}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|O_i - P_i|}{O_i}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(O_i - P_i)^2$$

$N$ = Number of evaluation samples
$O_i$ = Observed concentrations of particles
$P_i$ = Predicted concentration of $PM_{2.5}$ particles

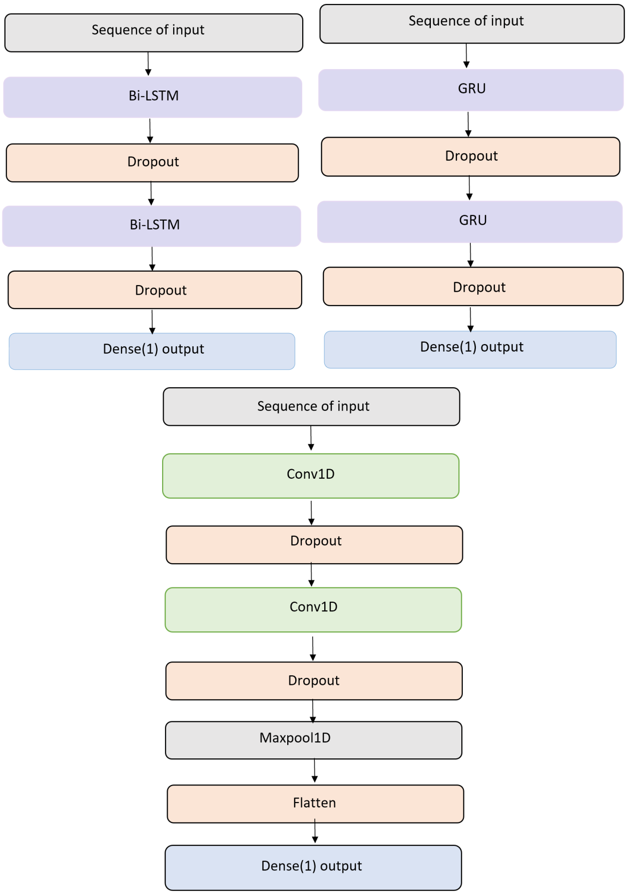


**Figure 2: The Bi-LSTM, GRU and TCN networks**

We design one deep Bi-LSTM and one GRU recurrent neural networks (Figure 2) following mostly the same parameters. Each of the two recurrent networks is stacked by 2 layers and finally one dense layer. We randomly and uniformly initialize the the kernel's weights using random uniform for both the Bi-LSTM and GRU models. We apply the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$, MAPE as the loss function and the adam algorithm as an optimizer. The TCN was built with one and later with two deep convolutional layers (Figure 2). Both convolutional layers have each a kernels size and a dilation rate that varies for each dataset. A max-pool is added after the second convolutional layer to help reduce some feature dependencies. After the max-pool, the outputs are flattened to pass into a dense layer with one neuron as our output.

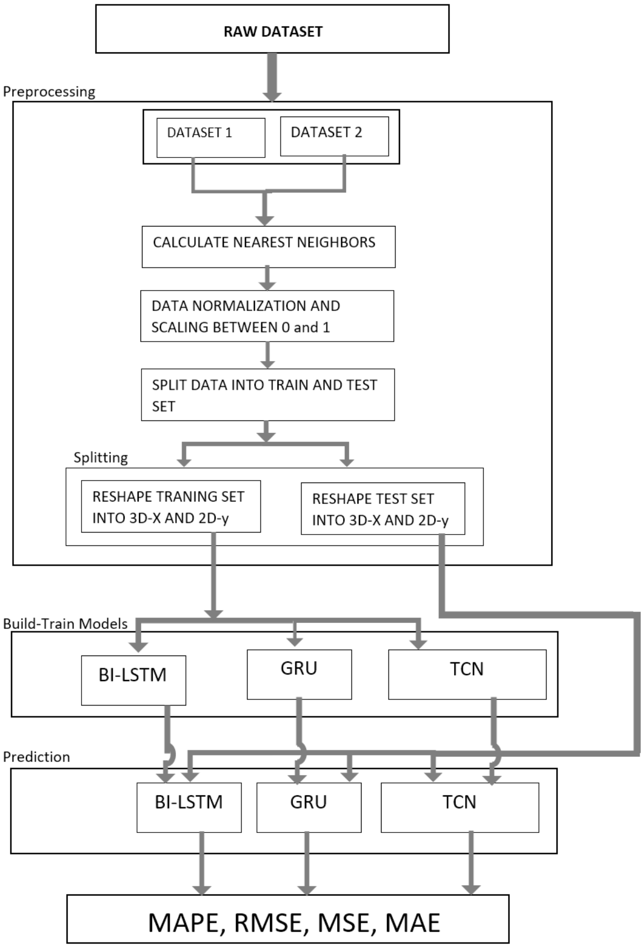

RAW DATASET
Preprocessing
DATASET 1
DATASET 2
CALCULATE NEAREST NEIGHBORS
DATA NORMALIZATION AND SCALING BETWEEN 0 and 1
SPLIT DATA INTO TRAIN AND TEST SET
Splitting
RESHAPE TRANING SET INTO 3D-X AND 2D-y
RESHAPE TEST SET INTO 3D-X AND 2D-y
Build-Train Models
BI-LSTM
GRU
TCN
Prediction
BI-LSTM
GRU
TCN
MAPE, RMSE, MSE, MAE

**Figure 3: Flowchart of the Model**

While training the neural networks, some of the models were overfitting as the performance on the training set was better than the validation set. To solve such problems, we add tested multiple techniques such as reducing the number of units, adding dropout and use the early stopping method which consists of terminating the training when the error reaches a certain threshold.

The framework of our model is shown in Figure 3. When implementing a neural network, it is important to normalize and scale the data to avoid overfitting. In our case, it is even more important to normalize and scale the data between 0 and 1 because we are working with outliers and sigmoid activation function.

We set up two types of experiments. The first type of experiments is to see if a neural network can improve based on the number of neighbors and the number of influencing days. For all neural networks implemented, each dataset was divided in three sets: 81% for training, 9% for validation, and 10% for testing. The second type of experiments is to compare the three neural networks in our model.

For all experiments, we implement them on Windows 10 with the processor "Intel(R) Core(TM) i7(R)-10510U CPU @ 1.80GHz" 16 GB DDR4 system memory, and GPU "2 GB NVIDIA GeForce MX250".

## 5 EXPERIMENT RESULTS

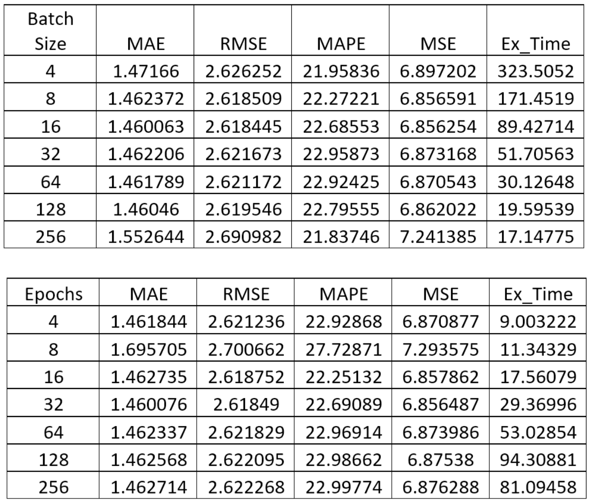
A neural network is not trained with all the training data all at once. Instead, the data is divided into fixed batches which are fed sequentially to the layers. Small batch sizes are proven to make the training loss converge faster in few epochs while bigger batches can be processed in parallel hence are computational efficient [5]. To choose the optimal batch size, we try several batch sizes and choose the optimal batch size based on MAPE.

During one epoch, all the *batches* are used only once to train the neural network. If a neural network is trained on small number of cycle, it will not learn enough. So, we might think that we should use more epochs to learn more which might be the solution. While it is common sense to think that the more you train the better you learn; but that's not the case for neural networks. A neural network which learns too much will overfit meaning that it will learn to perform perfectly on the training data and perform poorly on the validation or testing data. So we must choose an optimal number of epochs to avoid overfitting.

We first find the optimal number of batch sizes and the optimal number of epochs for each neural network in our model. Then we explore the influencing days and neighbors for each neural network in our model.

- *Bi-LSTM*:
  The batch table (upper of Figure 4) shows the relations between each batch size and the resulting MAPE. The lower of Figure 4 shows the result of training under different epochs. We complete more tests and some of our first choice parameters were changing drastically per test. So we always pick the three parameters that lead to the three smallest *MAPE*. After that, we complete the same tests again for three more times. The parameter with the highest probability of getting a low *MAPE* is chosen. The final Bi-LSTM parameters that we choose for training are the following:
  - Bi-LSTM Units per layer = 5
  - Batch size = 16
  - Number of epoch = 32
  - Number of Bi-LSTM hidden layers = 2
  - Number of Dense layer = 1
  - Dropout = 0.0

  We wanted to know how the neural network performs on different days. Figure 5 shows the number of days in the horizontal axis and the resulting neural network MAPE on the vertical axis. MAPE fluctuates but does not decrease when the number of days increases from 1 to 6. It means



| Batch Size | MAE | RMSE | MAPE | MSE | Ex_Time |
|---|---|---|---|---|---|
| 4 | 1.47166 | 2.626252 | 21.95836 | 6.897202 | 323.5052 |
| 8 | 1.462372 | 2.618509 | 22.27221 | 6.856591 | 171.4519 |
| 16 | 1.460063 | 2.618445 | 22.68553 | 6.856254 | 89.42714 |
| 32 | 1.462206 | 2.621673 | 22.95873 | 6.873168 | 51.70563 |
| 64 | 1.461789 | 2.621172 | 22.92425 | 6.870543 | 30.12648 |
| 128 | 1.46046 | 2.619546 | 22.79555 | 6.862022 | 19.59539 |
| 256 | 1.552644 | 2.690982 | 21.83746 | 7.241385 | 17.14775 |

| Epochs | MAE | RMSE | MAPE | MSE | Ex_Time |
|---|---|---|---|---|---|
| 4 | 1.461844 | 2.621236 | 22.92868 | 6.870877 | 9.003222 |
| 8 | 1.695705 | 2.700662 | 27.72871 | 7.293575 | 11.34329 |
| 16 | 1.462735 | 2.618752 | 22.25132 | 6.857862 | 17.56079 |
| 32 | 1.460076 | 2.61849 | 22.69089 | 6.856487 | 29.36996 |
| 64 | 1.462337 | 2.621829 | 22.96914 | 6.873986 | 53.02854 |
| 128 | 1.462568 | 2.622095 | 22.98662 | 6.87538 | 94.30881 |
| 256 | 1.462714 | 2.622268 | 22.99774 | 6.876288 | 81.09458 |

**Figure 4: Batch Size (upper) and Epochs (lower) for Bi-LSTM**

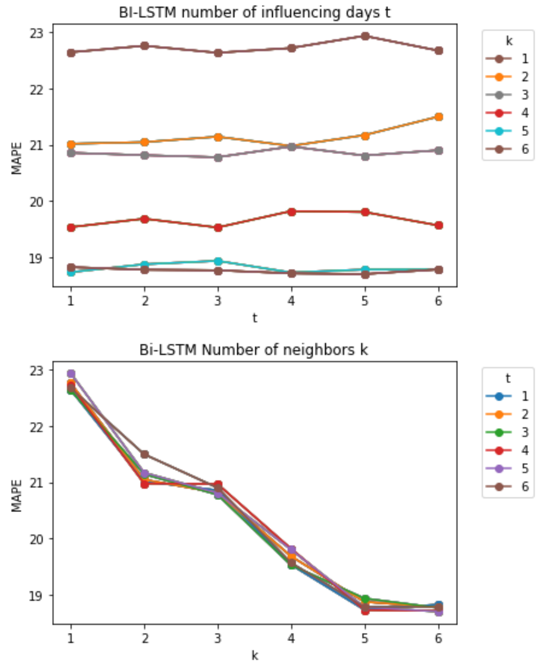


**Figure 5: Number of influencing days (upper) and number of neighbors (lower) for Bi-LSTM**

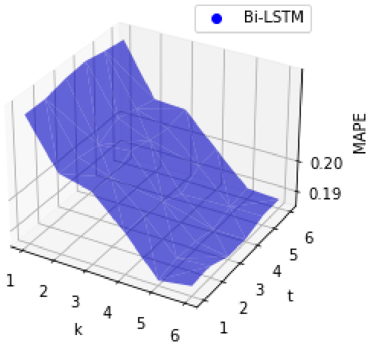
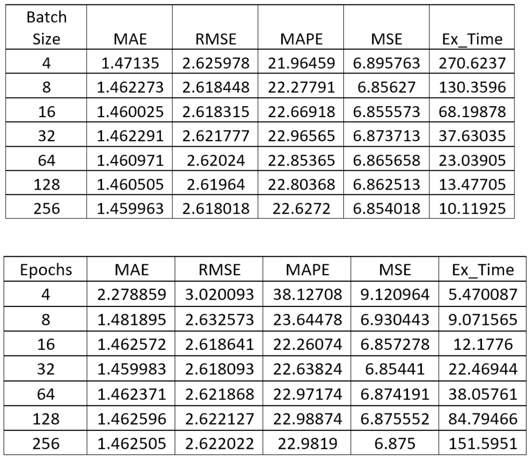


**Figure 6: Bi-LSTM 3D representation of t, k and MAPE**

that a small change in the number of days may not be very influential in the error calculation. In general, more days lead to a worse prediction because our Bi-LSTM model is a single prediction model which predicts one value for all stations. Such model makes the operations computationally efficient but it is subject to a known common error called Error Accumulation which increases the more days we try to predict [27]. Figure 5 shows that when we use more nearest neighbors, *MAPE* decreases. So, our Bi-LSTM can learn the correlation between the number of neighbors. Figure 6 is showing the relation between the number of days, the number of neighbors and the *MAPE*.

- *Gated Recurrent Unit:*
  The process of finding the batch size for the GRU model is the same as the Bi-LSTM. The batch table (upper of Figure 7) shows that MAPE is increasing with the increasing batch size. The epochs table (lower of Figure 7) shows that MAPE decreases with greater number of epochs. It means that the neural network is learning.



| Batch Size | MAE | RMSE | MAPE | MSE | Ex_Time |
|---|---|---|---|---|---|
| 4 | 1.47135 | 2.625978 | 21.96459 | 6.895763 | 270.6237 |
| 8 | 1.462273 | 2.618448 | 22.27791 | 6.85627 | 130.3596 |
| 16 | 1.460025 | 2.618315 | 22.66918 | 6.855573 | 68.19878 |
| 32 | 1.462291 | 2.621777 | 22.96565 | 6.873713 | 37.63035 |
| 64 | 1.460971 | 2.62024 | 22.85365 | 6.865658 | 23.03905 |
| 128 | 1.460505 | 2.61964 | 22.80368 | 6.862513 | 13.47705 |
| 256 | 1.459963 | 2.618018 | 22.6272 | 6.854018 | 10.11925 |

| Epochs | MAE | RMSE | MAPE | MSE | Ex_Time |
|---|---|---|---|---|---|
| 4 | 2.278859 | 3.020093 | 38.12708 | 9.120964 | 5.470087 |
| 8 | 1.481895 | 2.632573 | 23.64478 | 6.930443 | 9.071565 |
| 16 | 1.462572 | 2.618641 | 22.26074 | 6.857278 | 12.1776 |
| 32 | 1.459983 | 2.618093 | 22.63824 | 6.85441 | 22.46944 |
| 64 | 1.462371 | 2.621868 | 22.97174 | 6.874191 | 38.05761 |
| 128 | 1.462596 | 2.622127 | 22.98874 | 6.875552 | 84.79466 |
| 256 | 1.462505 | 2.622022 | 22.9819 | 6.875 | 151.5951 |

**Figure 7: Batch Size and Epoch for GRU**

After doing more tests, we pick the optimal parameters as follow:

- GRU Units per layer = 5
- Batch size = 16
- Number of epoch = 32
- Number of GRU hidden layers = 2
- Number of Dense layer = 1
- Dropout = 0.3

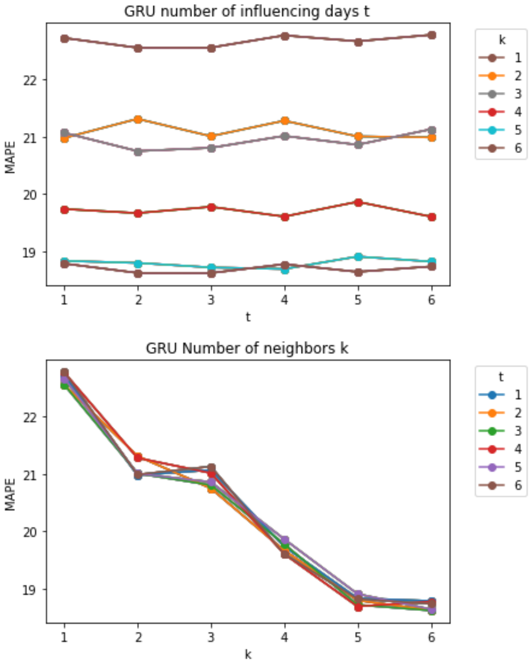


**Figure 8: Number of influencing days (upper) and number of neighbors (lower) for GRU**

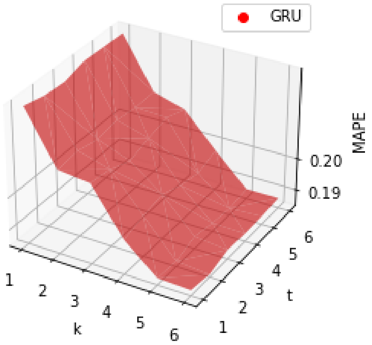
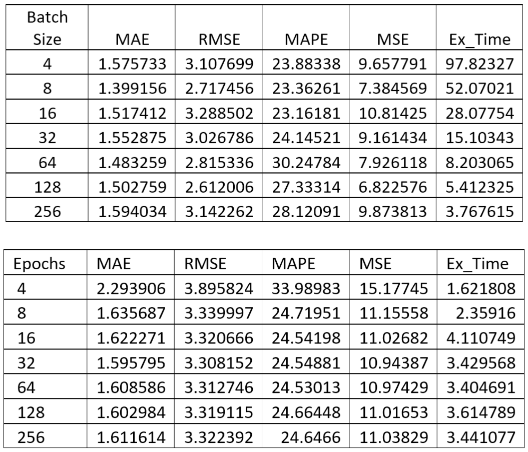


**Figure 9: GRU 3D representation of t, k and MAPE**

Figure 8 shows that the *MAPE* fluctuates when the number of days increases but the change is not significant. Figure 8 also shows that the more number of neighbors we add, the lower MAPE. So the number of neighbors influence the error significantly. Figure 9 is a 3D showing the relationship between the number of days, the number of neighbors and the resulting MAPE.

- *Temporal Convolution Neural Network:*
The way of finding the optimal parameters for the TCN is different than the Bi-LSTM and GRU models. TCN layers has filters instead of neuron units. Each layer output size is different than the input size because TCN unit has dilation rates which yields an output different than the input. So, the size of input should match the TCN units requirements. TCN input takes the following (number of samples, number of steps, number of features). We get an error our input data does not match the required input. We also need to manually adjust the hidden layers to match the output of each TCN layer.



| Batch Size | MAE | RMSE | MAPE | MSE | Ex_Time |
|---|---|---|---|---|---|
| 4 | 1.575733 | 3.107699 | 23.88338 | 9.657791 | 97.82327 |
| 8 | 1.399156 | 2.717456 | 23.36261 | 7.384569 | 52.07021 |
| 16 | 1.517412 | 3.288502 | 23.16181 | 10.81425 | 28.07754 |
| 32 | 1.552875 | 3.026786 | 24.14521 | 9.161434 | 15.10343 |
| 64 | 1.483259 | 2.815336 | 30.24784 | 7.926118 | 8.203065 |
| 128 | 1.502759 | 2.612006 | 27.33314 | 6.822576 | 5.412325 |
| 256 | 1.594034 | 3.142262 | 28.12091 | 9.873813 | 3.767615 |

| Epochs | MAE | RMSE | MAPE | MSE | Ex_Time |
|---|---|---|---|---|---|
| 4 | 2.293906 | 3.895824 | 33.98983 | 15.17745 | 1.621808 |
| 8 | 1.635687 | 3.339997 | 24.71951 | 11.15558 | 2.35916 |
| 16 | 1.622271 | 3.320666 | 24.54198 | 11.02682 | 4.110749 |
| 32 | 1.595795 | 3.308152 | 24.54881 | 10.94387 | 3.429568 |
| 64 | 1.608586 | 3.312746 | 24.53013 | 10.97429 | 3.404691 |
| 128 | 1.602984 | 3.319115 | 24.66448 | 11.01653 | 3.614789 |
| 256 | 1.611614 | 3.322392 | 24.6466 | 11.03829 | 3.441077 |

**Figure 10: Batch Size (upper) and Epochs (lower) for TCN**

The batch table (upper of Figure 10) shows different batch sizes that we tested. These data were obtained with $k = 1$ and $t = 1$ and kernel size =1. The epochs table (lower of Figure 10) shows the influence of different epochs on different errors. After multiples tests. We chose some parameters that could be used to train and test on all $k$ and $t$ as the following:

- TCN filters per layer = 5
- Batch size = 16
- Number of epoch = 32
- Number of TCN hidden layers = 1
- Number of Dense layer = 1
- Dropout = 0.0

Figure 11 shows barely no change to the increase of days. So the number of days either has no influence on the TCN or the increase in days is too small for the network to learn something. Figure 11 also shows that TCN notices the change in nearest neighbors, MAPE go straight down and up instead of keep on going down. The TCN either overestimates or underestimate the impact of nearest neighbors on the calculation of MAPE. Figure 12 shows the relation between $t$, $k$, and the resulting MAPE. We can see some sharp changes when the number of neighbors change but the changes are not coherent.

Figure 13, Figure 14, and Figure 15 show all the 36 measurements for the Bi-LSTM, GRU, and TCN, respectively. By comparing their MAPE, we can see that the Bi-LSTM model starts with the lowest *MAPE* followed by the GRU. When we compare the execution time of the three, we can see that the TCN leads the other two and then is followed by the GRU. By comparing the rate of decrease of the *MAPE*, we see that the GRU is the fastest followed by the Bi-LSTM. The GRU is the model that ends with the lowest *MAPE*.

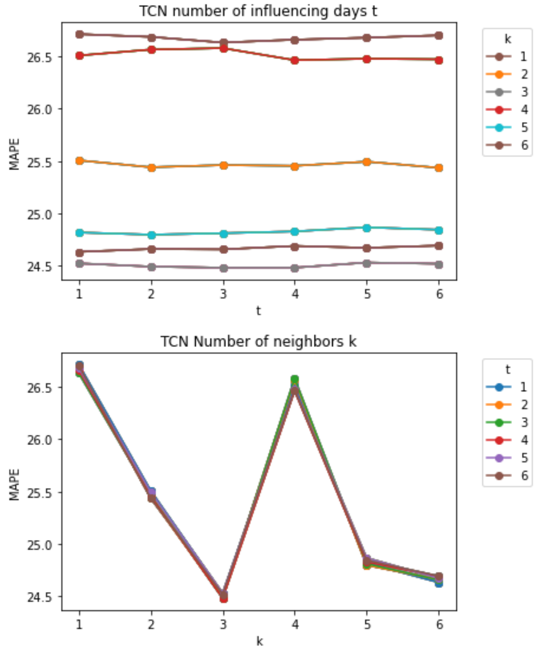


**Figure 11: TCN Number of influencing days (upper) and number of neighbors (lower) for TCN**

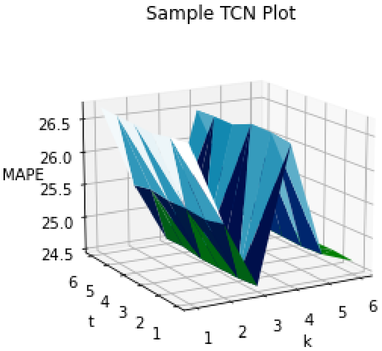
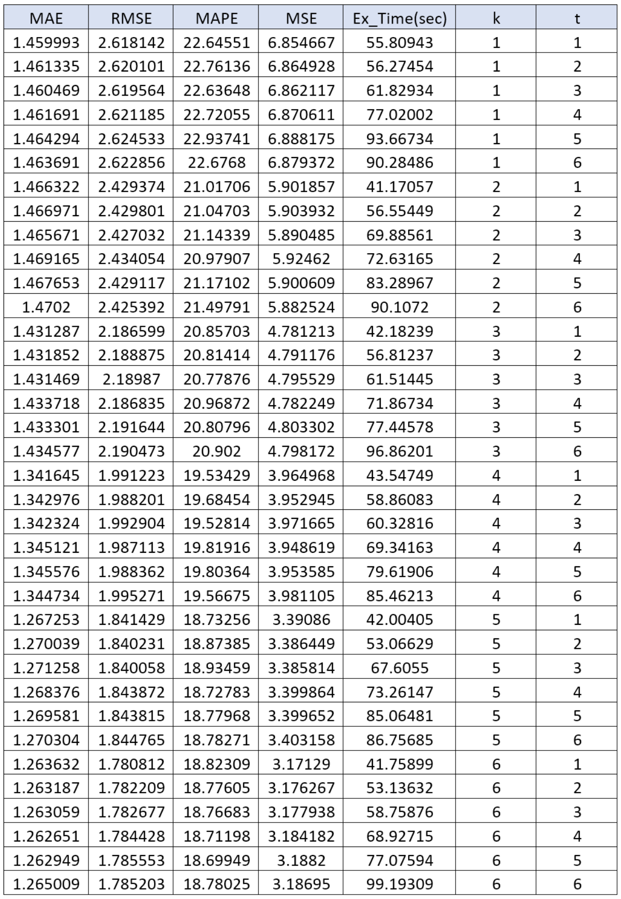


**Figure 12: TCN 3D representation of t, k and MAPE**

We then apply the k-d tree algorithm to find the nearest neighbors from 1 to 6. After that, we build three neural networks models and test the influence of spatiotemporal correlation on the estimation of MAPE. Once the testing done, we analyze the outputs to see the patterns in each result. Our proposed preprocessing technique was to help reduce the overfitting of the models and allow them to easily find the patterns which can lead to better estimations. However the results show that increasing the dataset size with dummy data also increases the error estimation and it does not effectively solve the overfitting problem. From all the observations above, we conclude that Bi-LSTM is optimal for predicting $PM_{2.5}$ when spatiotemporal data is not considered. Once the spatiotemporal parameters get included, we see that the GRU model takes over and performs better than the Bi-LSTM. GRU is also faster than the Bi-LSTM. Finally, the TCN gets the least accuracy but it is the fastest of the three neural networks models. Unfortunately, it is the model which could not clearly extract the spatial correlation in the data. This is due to its lack of recurrence. Between the three models, we choose the GRU because its runtime is multiples times faster that the Bi-LSTM and its accuracy improves faster while extracting spatiotemporal correlation in the data.

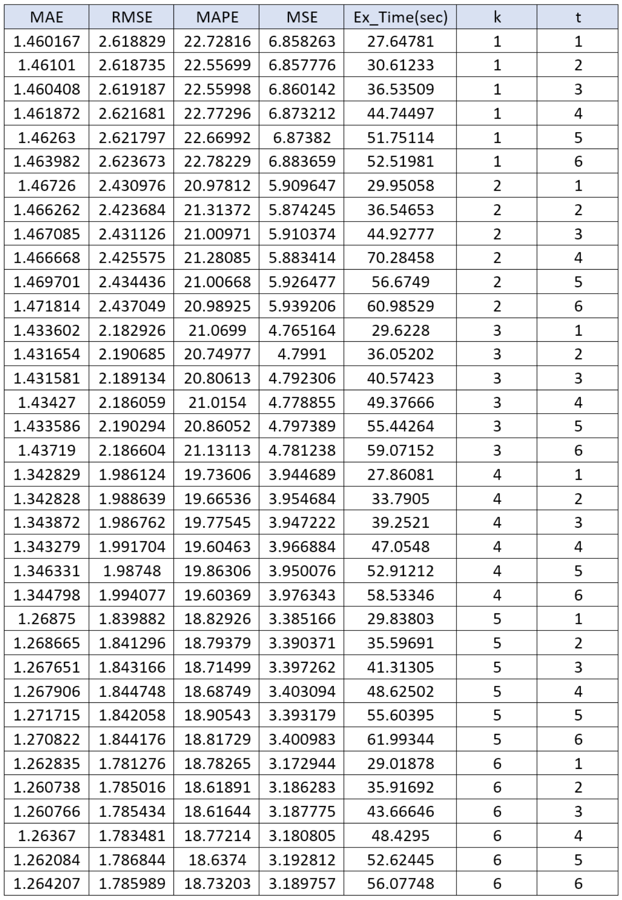


| MAE | RMSE | MAPE | MSE | Ex_Time(sec) | k | t |
|---|---|---|---|---|---|---|
| 1.459993 | 2.618142 | 22.64551 | 6.854667 | 55.80943 | 1 | 1 |
| 1.461335 | 2.620101 | 22.76136 | 6.864928 | 56.27454 | 1 | 2 |
| 1.460469 | 2.619564 | 22.63648 | 6.862117 | 61.82934 | 1 | 3 |
| 1.461691 | 2.621185 | 22.72055 | 6.870611 | 77.02002 | 1 | 4 |
| 1.464294 | 2.624533 | 22.93741 | 6.888175 | 93.66734 | 1 | 5 |
| 1.463691 | 2.622856 | 22.6768 | 6.879372 | 90.28486 | 1 | 6 |
| 1.466322 | 2.429374 | 21.01706 | 5.901857 | 41.17057 | 2 | 1 |
| 1.466971 | 2.429801 | 21.04703 | 5.903932 | 56.55449 | 2 | 2 |
| 1.465671 | 2.427032 | 21.14339 | 5.890485 | 69.88561 | 2 | 3 |
| 1.469165 | 2.434054 | 20.97907 | 5.92462 | 72.63165 | 2 | 4 |
| 1.467653 | 2.429117 | 21.17102 | 5.900609 | 83.28967 | 2 | 5 |
| 1.4702 | 2.425392 | 21.49791 | 5.882524 | 90.1072 | 2 | 6 |
| 1.431287 | 2.186599 | 20.85703 | 4.781213 | 42.18239 | 3 | 1 |
| 1.431852 | 2.188875 | 20.81414 | 4.791176 | 56.81237 | 3 | 2 |
| 1.431469 | 2.18987 | 20.77876 | 4.795529 | 61.51445 | 3 | 3 |
| 1.433718 | 2.186835 | 20.96872 | 4.782249 | 71.86734 | 3 | 4 |
| 1.433301 | 2.191644 | 20.80796 | 4.803302 | 77.44578 | 3 | 5 |
| 1.434577 | 2.190473 | 20.902 | 4.798172 | 96.86201 | 3 | 6 |
| 1.341645 | 1.991223 | 19.53429 | 3.964968 | 43.54749 | 4 | 1 |
| 1.342976 | 1.988201 | 19.68454 | 3.952945 | 58.86083 | 4 | 2 |
| 1.342324 | 1.992904 | 19.52814 | 3.971665 | 60.32816 | 4 | 3 |
| 1.345121 | 1.987113 | 19.81916 | 3.948619 | 69.34163 | 4 | 4 |
| 1.345576 | 1.988362 | 19.80364 | 3.953585 | 79.61906 | 4 | 5 |
| 1.344734 | 1.995271 | 19.56675 | 3.981105 | 85.46213 | 4 | 6 |
| 1.267253 | 1.841429 | 18.73256 | 3.39086 | 42.00405 | 5 | 1 |
| 1.270039 | 1.840231 | 18.87385 | 3.386449 | 53.06629 | 5 | 2 |
| 1.271258 | 1.840058 | 18.93459 | 3.385814 | 67.6055 | 5 | 3 |
| 1.268376 | 1.843872 | 18.72783 | 3.399864 | 73.26147 | 5 | 4 |
| 1.269581 | 1.843815 | 18.77968 | 3.399652 | 85.06481 | 5 | 5 |
| 1.270304 | 1.844765 | 18.78271 | 3.403158 | 86.75685 | 5 | 6 |
| 1.263632 | 1.780812 | 18.82309 | 3.17129 | 41.75899 | 6 | 1 |
| 1.263187 | 1.782209 | 18.77605 | 3.176267 | 53.13632 | 6 | 2 |
| 1.263059 | 1.782677 | 18.76683 | 3.177938 | 58.75876 | 6 | 3 |
| 1.262651 | 1.784428 | 18.71198 | 3.184182 | 68.92715 | 6 | 4 |
| 1.262949 | 1.785553 | 18.69949 | 3.1882 | 77.07594 | 6 | 5 |
| 1.265009 | 1.785203 | 18.78025 | 3.18695 | 99.19309 | 6 | 6 |

**Figure 13: Measurements for Bi-LSTM**

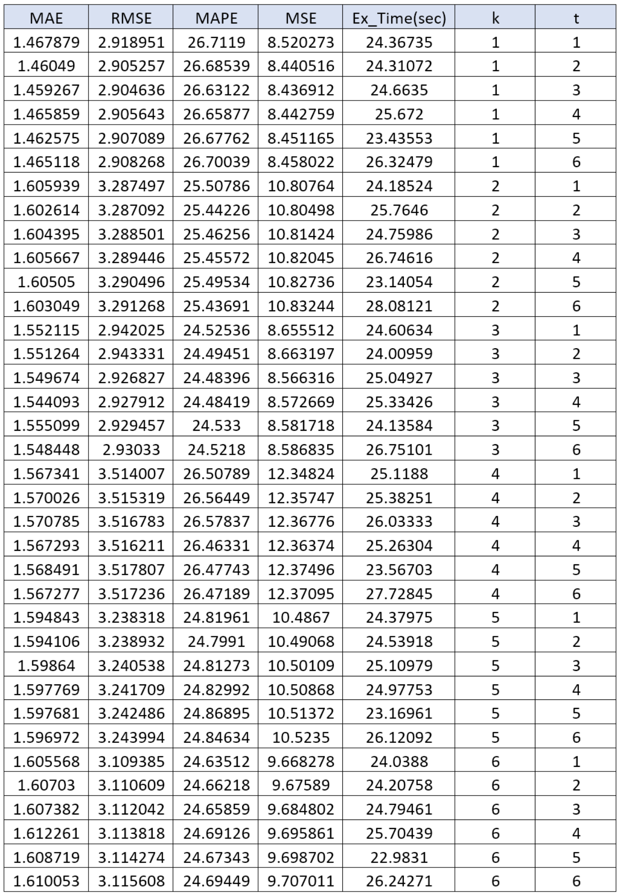
## 6 CONCLUSION

We build three neural networks models and use them to estimate the concentration of $PM_{2.5}$. We experimented two data augmentation techniques. Our results prove that using the spatiotemporal technique proposed by Li et al. [25], yields better results when we used more nearest neighbors. We are able to verify that the Bi-LSTM model yields better results when time and data correlation are not taken in account. We also show that the GRU model almost as accurate as the Bi-LSTM and it even outperforms the Bi-LSTM when the execution time and the nearest neighbors are taken in account. Finally, showed that even if TCN is the least accurate of all three, it is also the fastest of the three models. We think that TCN, used as a sequential model should not be used to extract multiple spatial correlation features.



| MAE | RMSE | MAPE | MSE | Ex_Time(sec) | k | t |
|---|---|---|---|---|---|---|
| 1.460167 | 2.618829 | 22.72816 | 6.858263 | 27.64781 | 1 | 1 |
| 1.46101 | 2.618735 | 22.55699 | 6.857776 | 30.61233 | 1 | 2 |
| 1.460408 | 2.619187 | 22.55998 | 6.860142 | 36.53509 | 1 | 3 |
| 1.461872 | 2.621681 | 22.77296 | 6.873212 | 44.74497 | 1 | 4 |
| 1.46263 | 2.621797 | 22.66992 | 6.87382 | 51.75114 | 1 | 5 |
| 1.463982 | 2.623673 | 22.78229 | 6.883659 | 52.51981 | 1 | 6 |
| 1.46726 | 2.430976 | 20.97812 | 5.909647 | 29.95058 | 2 | 1 |
| 1.466262 | 2.423684 | 21.31372 | 5.874245 | 36.54653 | 2 | 2 |
| 1.467085 | 2.431126 | 21.00971 | 5.910374 | 44.92777 | 2 | 3 |
| 1.466668 | 2.425575 | 21.28085 | 5.883414 | 70.28458 | 2 | 4 |
| 1.469701 | 2.434436 | 21.00668 | 5.926477 | 56.6749 | 2 | 5 |
| 1.471814 | 2.437049 | 20.98925 | 5.939206 | 60.98529 | 2 | 6 |
| 1.433602 | 2.182926 | 21.0699 | 4.765164 | 29.6228 | 3 | 1 |
| 1.431654 | 2.190685 | 20.74977 | 4.7991 | 36.05202 | 3 | 2 |
| 1.431581 | 2.189134 | 20.80613 | 4.792306 | 40.57423 | 3 | 3 |
| 1.43427 | 2.186059 | 21.0154 | 4.778855 | 49.37666 | 3 | 4 |
| 1.433586 | 2.190294 | 20.86052 | 4.797389 | 55.44264 | 3 | 5 |
| 1.43719 | 2.186604 | 21.13113 | 4.781238 | 59.07152 | 3 | 6 |
| 1.342829 | 1.986124 | 19.73606 | 3.944689 | 27.86081 | 4 | 1 |
| 1.342828 | 1.988639 | 19.66536 | 3.954684 | 33.7905 | 4 | 2 |
| 1.343872 | 1.986762 | 19.77545 | 3.947222 | 39.2521 | 4 | 3 |
| 1.343279 | 1.991704 | 19.60463 | 3.966884 | 47.0548 | 4 | 4 |
| 1.346331 | 1.98748 | 19.86306 | 3.950076 | 52.91212 | 4 | 5 |
| 1.344798 | 1.994077 | 19.60369 | 3.976343 | 58.53346 | 4 | 6 |
| 1.26875 | 1.839882 | 18.82926 | 3.385166 | 29.83803 | 5 | 1 |
| 1.268665 | 1.841296 | 18.79379 | 3.390371 | 35.59691 | 5 | 2 |
| 1.267651 | 1.843166 | 18.71499 | 3.397262 | 41.31305 | 5 | 3 |
| 1.267906 | 1.844748 | 18.68749 | 3.403094 | 48.62502 | 5 | 4 |
| 1.271715 | 1.842058 | 18.90543 | 3.393179 | 55.60395 | 5 | 5 |
| 1.270822 | 1.844176 | 18.81729 | 3.400983 | 61.99344 | 5 | 6 |
| 1.262835 | 1.781276 | 18.78265 | 3.172944 | 29.01878 | 6 | 1 |
| 1.260738 | 1.785016 | 18.61891 | 3.186283 | 35.91692 | 6 | 2 |
| 1.260766 | 1.785434 | 18.61644 | 3.187775 | 43.66646 | 6 | 3 |
| 1.26367 | 1.783481 | 18.77214 | 3.180805 | 48.4295 | 6 | 4 |
| 1.262084 | 1.786844 | 18.6374 | 3.192812 | 52.62445 | 6 | 5 |
| 1.264207 | 1.785989 | 18.73203 | 3.189757 | 56.07748 | 6 | 6 |

**Figure 14: Measurements for GRU**



| MAE | RMSE | MAPE | MSE | Ex_Time(sec) | k | t |
|---|---|---|---|---|---|---|
| 1.467879 | 2.918951 | 26.7119 | 8.520273 | 24.36735 | 1 | 1 |
| 1.46049 | 2.905257 | 26.68539 | 8.440516 | 24.31072 | 1 | 2 |
| 1.459267 | 2.904636 | 26.63122 | 8.436912 | 24.6635 | 1 | 3 |
| 1.465859 | 2.905643 | 26.65877 | 8.442759 | 25.672 | 1 | 4 |
| 1.462575 | 2.907089 | 26.67762 | 8.451165 | 23.43553 | 1 | 5 |
| 1.465118 | 2.908268 | 26.70039 | 8.458022 | 26.32479 | 1 | 6 |
| 1.605939 | 3.287497 | 25.50786 | 10.80764 | 24.18524 | 2 | 1 |
| 1.602614 | 3.287092 | 25.44226 | 10.80498 | 25.7646 | 2 | 2 |
| 1.604395 | 3.288501 | 25.46256 | 10.81424 | 24.75986 | 2 | 3 |
| 1.605667 | 3.289446 | 25.45572 | 10.82045 | 26.74616 | 2 | 4 |
| 1.60505 | 3.290496 | 25.49534 | 10.82736 | 23.14054 | 2 | 5 |
| 1.603049 | 3.291268 | 25.43691 | 10.83244 | 28.08121 | 2 | 6 |
| 1.552115 | 2.942025 | 24.52536 | 8.655512 | 24.60634 | 3 | 1 |
| 1.551264 | 2.943331 | 24.49451 | 8.663197 | 24.00959 | 3 | 2 |
| 1.549674 | 2.926827 | 24.48396 | 8.566316 | 25.04927 | 3 | 3 |
| 1.544093 | 2.927912 | 24.48419 | 8.572669 | 25.33426 | 3 | 4 |
| 1.555099 | 2.929457 | 24.533 | 8.581718 | 24.13584 | 3 | 5 |
| 1.548448 | 2.93033 | 24.5218 | 8.586835 | 26.75101 | 3 | 6 |
| 1.567341 | 3.514007 | 26.50789 | 12.34824 | 25.1188 | 4 | 1 |
| 1.570026 | 3.515319 | 26.56449 | 12.35747 | 25.38251 | 4 | 2 |
| 1.570785 | 3.516783 | 26.57837 | 12.36776 | 26.03333 | 4 | 3 |
| 1.567293 | 3.516211 | 26.46331 | 12.36374 | 25.26304 | 4 | 4 |
| 1.568491 | 3.517807 | 26.47743 | 12.37496 | 23.56703 | 4 | 5 |
| 1.567277 | 3.517236 | 26.47189 | 12.37095 | 27.72845 | 4 | 6 |
| 1.594843 | 3.238318 | 24.81961 | 10.4867 | 24.37975 | 5 | 1 |
| 1.594106 | 3.238932 | 24.7991 | 10.49068 | 24.53918 | 5 | 2 |
| 1.59864 | 3.240538 | 24.81273 | 10.50109 | 25.10979 | 5 | 3 |
| 1.597769 | 3.241709 | 24.82992 | 10.50868 | 24.97753 | 5 | 4 |
| 1.597681 | 3.242486 | 24.86895 | 10.51372 | 23.16961 | 5 | 5 |
| 1.596972 | 3.243994 | 24.84634 | 10.5235 | 26.12092 | 5 | 6 |
| 1.605568 | 3.109385 | 24.63512 | 9.668278 | 24.0388 | 6 | 1 |
| 1.60703 | 3.110609 | 24.66218 | 9.67589 | 24.20758 | 6 | 2 |
| 1.607382 | 3.112042 | 24.65859 | 9.684802 | 24.79461 | 6 | 3 |
| 1.612261 | 3.113818 | 24.69126 | 9.695861 | 25.70439 | 6 | 4 |
| 1.608719 | 3.114274 | 24.67343 | 9.698702 | 22.9831 | 6 | 5 |
| 1.610053 | 3.115608 | 24.69449 | 9.707011 | 26.24271 | 6 | 6 |

**Figure 15: Measurements for TCN**

In the future, we think that using an ensemble neural networks to build hybrid neural networks and adding more features such as temperature, wind and other pollutant particles can increase the stability and accuracy of our model. We could also experiment on the accuracy of estimating one location at a time other multiples locations at once. Using an interpolation technique such as linear interpolation could have given better results. Finally, we could test the efficiency of the model in multivariate domains such as traffic data combined with meteorological data which could help better understand humans impact on the climate change.

## REFERENCES

[1] US Environmental Protection Agency and William K. Reilly. 1990. State Implementation Plans; General Preamble for the Implementation of Title I of the Clean Air Act Amendments of 1990. (Nov. 1990), 74 pages. https://www.epa.gov/criteria-air-pollutants/naaqs-table

[2] Saad Albawi, Tareq Abed Mohammed, and Saad ALZAWI. 2017. Understanding of a Convolutional Neural Network. https://doi.org/10.1109/ICEngTechnol.2017.8308186

[3] Preeti Bamane and Mangal Patil. 2020. INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY A comparative study of different LSTM neural networks in predicting air pollutant concentrations. *Indian Journal of Science and Technology* 13 (11 2020), 3664. https://doi.org/10.17485/IJST/v13i35.1276

[4] Yitian Chen, K Yanfei, C Yixiong, and W Zizhuo. 2020. Probabilistic Forecasting with Temporal Convolutional Neural Network. *Neurocomputing* 399 (03 2020). https://doi.org/10.1016/j.neucom.2020.03.011

[5] Aditya Devarakonda, Maxim Naumov, and Michael Garland. 2018. AdaBatch: Adaptive Batch Sizes for Training Deep Neural Networks. arXiv:1712.02029 [cs.LG]

[6] Commissioner Howard Zucker, M.D. 2018. Fine Particles (PM 2.5) Questions and Answers. *NY State Department of Health* 1, 1 (2018), 1–3. https://www.health.ny.gov/environmental/indoors/air/pmq_a.htm

[7] Chiou-Jye Huang and Ping-Huan Kuo. 2018. A Deep CNN-LSTM Model for Particulate Matter (PM2.5) Forecasting in Smart Cities. *Sensors* 18 (07 2018), 2220. https://doi.org/10.3390/s18072220

[8] Guirguis Karim, S Christoph, G Andre, and Co. 2020. SELD-TCN: Sound Event Localization & Detection via Temporal Convolutional Networks. https://doi.org/10.23919/Eusipco47968.2020.9287716

[9] Colin Lea, Michael Flynn, Rene Vidal, Austin Reiter, and Gregory Hager. 2017. Temporal Convolutional Networks for Action Segmentation and Detection. 1003–1012. https://doi.org/10.1109/CVPR.2017.113

[10] Colin Lea, V René, R Austin, and H Gregory. 2016. Temporal Convolutional Networks: A Unified Approach to Action Segmentation. https://doi.org/10.1007/978-3-319-49409-8_7

[11] Lixin Li, Travis Losser, Charles Yorke, and Reinhard Piltner. 2014. Fast Inverse Distance Weighting-Based Spatiotemporal Interpolation: A Web-Based Application of Interpolating Daily Fine Particulate Matter PM2.5 in the Contiguous U.S. Using Parallel Programming and k-d Tree. *International journal of environmental research and public health* 11 (09 2014), 9101–9141. https://doi.org/10.3390/ijerph110909101

[12] Lixin Li and Peter Revesz. 2002. A Comparison of Spatio-temporal Interpolation Methods. 145–160. https://doi.org/10.1007/3-540-45799-2_11

[13] Lixin Li and Peter Revesz. 2004. Interpolation methods for spatio-temporal geographic data. (2004), 27 pages. https://doi.org/10.1016/S0198-9715(03)00018-8

[14] Lixin Li, Jie Tian, Xingyou Zhang, James Holt, and Reinhard Piltner. 2015. Estimating Population Exposure to Fine Particulate Matter in the Conterminous U.S.

using Shape Function-based Spatiotemporal Interpolation Method: A County Level Analysis. *GSTF international journal on computing* 1 (09 2015), 24–30.

[15] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Graph Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. (07 2017).

[16] Xianglong Luo, Danyang Li, Yu Yang, and Shengrui Zhang. 2019. Spatiotemporal traffic flow prediction with KNN and LSTM. *Journal of Advanced Transportation* 2019 (02 2019), 1–10. https://doi.org/10.1155/2019/4145353

[17] Navin Kumar Manaswi. 2018. Deep Learning with Applications Using Python. (2018), 219–227.

[18] Cromley Merwin David and Co. 2013. A Neural Network-based Method for Solving "Nested Hierarchy" Areal Interpolation Problems. *Cartography and Geographic Information Science* 36 (03 2013), 347–365. https://doi.org/10.1559/152304009789786335

[19] Weibiao Qiao, Tian Wencai, Yu Tian, Quan Yang, Yining Wang, and Jianzhuang Zhang. 2019. The Forecasting of PM2.5 Using a Hybrid Model Based on Wavelet Transform and an Improved Deep Learning Algorithm. *IEEE Access* PP (09 2019), 1–1. https://doi.org/10.1109/ACCESS.2019.2944755

[20] Dario Rethage, Jordi Pons, and Xavier Serra. 2018. A Wavenet for Speech Denoising. 5069–5073. https://doi.org/10.1109/ICASSP.2018.8462417

[21] Peter Revesz. 2014. *Spatiotemporal Interpolation Algorithms*. 1–5. https://doi.org/10.1007/978-1-4899-7993-3_803-2

[22] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. 2017. Deep Network Flow for Multi-Object Tracking. (06 2017).

[23] Mike Schuster and Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on* 45 (12 1997), 2673 – 2681. https://doi.org/10.1109/78.650093

[24] John D. Spengler and Ken Sexton. 1983. Indoor air pollution: a public health perspective. *Web of Science.* 221, 4605, Article 336 (July 1983), 9 pages. https://doi.org/10.1145/3161192

[25] Weitian Tong, Lixin Li, Xiaolu Zhou, Andrew Hamilton, and Kai Zhang. 2019. Deep learning PM2.5 concentrations with bidirectional LSTM RNN. *Air Quality, Atmosphere & Health* 12 (04 2019), 1–13. https://doi.org/10.1007/s11869-018-0647-4

[26] J. Jason West and Co. 2016. What We Breathe Impacts Our Health: Improving Understanding of the Link between Air Pollution and Health. *Environmental Science & Technology 50 (10), 4895-4904* (2016). https://doi.org/10.1021/acs.est.5b03827

[27] Yanlai Zhou, Fi-John Chang, Li-Chiu Chang, I-Feng Kao, and Yi-Shin Wang. 2018. Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts. *Journal of Cleaner Production* 209 (10 2018). https://doi.org/10.1016/j.jclepro.2018.10.243

# Q-Eclat: Vertical Mining of Interesting Quantitative Patterns

Thomas J. Czubryt
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Carson K. Leung*
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
Carson.Leung@UManitoba.ca

Adam G.M. Pazdor
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada

## ABSTRACT

Frequent pattern mining is a popular technique in big data mining and analytics. It discovers frequently occurring sets of items (e.g., popular merchandise items, frequently co-occurring events) from big data found in numerous database engineered applications. These frequent patterns can be discovered horizontally by transaction-centric mining algorithms or vertically by item-centric mining algorithms. Regardless of their mining direction (horizontal or vertical), traditional frequent pattern mining algorithms aim to discover Boolean frequent patterns in the sense that patterns capture the presence (or absence) of items within the discovered patterns. However, there are many real-life situations, in which quantities of items within the patterns are important. For example, the quantity of items may also affect profits of selling the items within the discovered patterns. Hence, in this paper, we present an algorithm for vertical mining of interesting quantitative frequent patterns. This Q-Eclat algorithm first represents the big data as a collection of equivalence classes according to their prefix item labels. Each domain item is represented by one of these classes. Their corresponding item-centric sets capture (a) IDs of transactions containing the item, as well as (b) the quantity of that item in each transaction. With this representation, our algorithm then vertically mines quantitative frequent patterns. When compared the existing MQA-M algorithm (which was built for quantitative horizontal frequent pattern mining), evaluation results show that our quantitative vertical Q-Eclat algorithm takes shorter runtime to mine quantitative frequent patterns.

## CCS CONCEPTS

• **Information systems** → **Data mining**; **Association rules**.

## KEYWORDS

Database engineered application, Frequent pattern mining, Quantitative data mining, Vertical pattern mining, Eclat, Transaction ID, Set

**ACM Reference Format:**
Thomas J. Czubryt, Carson K. Leung, and Adam G.M. Pazdor. 2022. Q-Eclat: Vertical Mining of Interesting Quantitative Patterns. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3548785.3548808

## 1 INTRODUCTION

Nowadays, big data [1, 2] can be found in numerous database engineered applications. With advances in technology, high volumes of a wide variety of data (which may be of different levels of varsity) are generated and collected at a high velocity for numerous real-life applications and services such as:

- medical/healthcare informatics [3–7] and disease analytics [8–12];
- transportation analytics [13–15];
- business analytics [16–20]; as well as
- social media mining [21–23] and social network analysis [24–28].

Embedded in these big data is implicit, previously unknown and potentially useful information and knowledge. This calls for big data management [29–31], big data mining [32–35], big data analytics [36, 37], as well as big data visualization and analytics [38–41].

*Association rule mining* is a popular technique for big data mining and analytics. It discovers rules that reveal interesting associations among the antecedents and consequences of the rules. Generally, these rules are mined by first discovering frequent patterns and then using these discovered frequent patterns to form the rules. *Frequent pattern mining* [42–45] aims to discover frequently occurring sets of items (aka **itemsets**)—such as popular merchandise items in shopping carts or shopper market baskets, or frequently co-located conferences/events—from big data. Given a series of transactions containing a set of items, frequent pattern mining seeks to determine the sets of items, which occur in a large number of transactions. In addition, we wish to discover interesting association rules. Association rules state that whenever a certain set of items occurs in a transaction, another set of items tends to occur in that transaction. The problems of frequent pattern mining and association rule mining form the basis of many real-life applications such as marketing in business, discovering biological patterns, studying human populations, web log mining, and many database engineered applications. Frequent pattern mining has been extended to the mining of other patterns such as network and graph mining [46–48], stream mining [49, 50], uncertain pattern mining [51–55], and utility pattern mining [56].

Frequent patterns can be discovered horizontally by transaction-centric mining algorithms or vertically by item-centric mining algorithms [57–62]. The Apriori algorithm [63, 64] is an example of

[*] Corresponding author: Carson.Leung@UManitoba.ca (C.K. Leung)

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548808

horizontal transaction-centric frequent pattern mining algorithms, with which data are represented as a collection of transactions. Each transaction captures the presence or absence of items. Alternatively, frequent patterns can also be discovered vertically. The Eclat (Equivalence CLAss Transformation) algorithm [61] is an example of vertical item-centric frequent pattern mining algorithms, with which data are represented as a collection of equivalence classes according to their prefix item labels. Each domain item is represented by one of these classes, and the corresponding transaction ID set for an item captures which transactions contain the specific item. Specifically, the set contains transaction IDs. An advantage of such a set representation is that the size of set is proportional to the density of the data. Sparse data would lead to a small transaction ID set. The algorithm was shown to be efficient as it takes advantage of set operations in the mining process.

Whether to mine frequent patterns horizontally or vertically, traditional algorithms aim to discover Boolean frequent patterns in the sense that patterns capture the presence (or absence) of items within the discovered patterns. While traditional frequent pattern mining and association rule mining are useful in many contexts, they have a major limitation. This limitation is that in traditional frequent pattern mining, we assume that every transaction either contains an item or does not contain the item. In other words, an item is contained in a transaction 0 or 1 times. For this reason, we can also refer to traditional frequent pattern mining as *Boolean frequent pattern mining*. However, in many real-world scenarios, a transaction can contain an item more than one time. For example, a person at a grocery store may buy multiple apples. To address this shortcoming, the notion of quantitative association rule mining or quantitative frequent pattern mining [65, 66] was introduced. *Quantitative frequent pattern mining* is essentially an extension of frequent pattern mining to allow transactions to contain an item more than once. Rather than just trying to find items (which commonly occur in transactions), there is a demand for discovering commonly occurring quantities of items. For example, in Boolean frequent pattern mining, one may discover that bananas are a frequently purchased item. In quantitative frequent pattern mining, one may discover that customers frequently purchase at least five bananas at a time. As another example, the quantity of items may also affect profits of selling the items within the discovered patterns.

Through the discovery of quantitative frequent patterns and quantitative association rules, we can obtain more interesting results than we would if Boolean association rule mining were used instead. In addition to receiving information about which items commonly occur together in transactions, we also obtain information regarding how many of each of those items tend to occur in transactions. MQA-M algorithm [65] extends the Apriori algorithm for mining quantitative association rules with multiple comparison operators—i.e., mining *quantitative frequent patterns* (aka sets of item expressions, i.e., *itemexpsets* for short)—horizontally.

We present in this paper we present a vertical equivalence class-based algorithm to mine quantitative frequent patterns (i.e., itemexpsets) vertically. The resulting Q-Eclat algorithm first represents the big data as a collection of bitmaps. Each item-centric transaction ID set captures the IDs of transactions containing the item, as well as the quantity of that item in each transaction. With this representation, our algorithm then vertically mines quantitative frequent patterns. When compared the existing MQA-M algorithm (which was built for quantitative frequent pattern mining), evaluation results show that our quantitative vertical Q-Eclat algorithm requires shorter execution time to mine frequent patterns. Our key contributions in this paper include our Q-Eclat algorithm and its pruning rules.

We organize the remainder of this paper as follows. We begin by presenting the mathematical framework for quantitative frequent pattern mining in Section 2. We discuss previously used algorithms of interest, such as the Apriori, Eclat, and MQA-M algorithms. Then, we formally introduce our Q-Eclat algorithms in Section 3. Pseudocode and an example are provided for the algorithm. Section 4 contains analysis of the algorithm and evaluation to compare our Q-Eclat with related works. Finally, we conclude in Section 5.

## 2 BACKGROUND AND RELATED WORKS

Here, we formally define quantitative frequent patterns and review relevant algorithms before describing our algorithm for quantitative frequent pattern mining.

### 2.1 Horizontal Boolean Frequent Pattern Mining with the Apriori Algorithm

As a common algorithm used in Boolean frequent pattern mining, the Apriori algorithm [64] provides the foundation for the MQA-M algorithm used for quantitative frequent pattern mining. The Apriori algorithm works by finding frequent patterns containing one item (i.e., 1-itemsets) first, and then finding patterns of higher cardinality (i.e., $k$-itemsets for $k \geq 2$) as the algorithm runs.

To elaborate, for any positive integer $k$, let $C_k$ be the set of candidate patterns containing $k$ items (i.e., *candidate $k$-itemsets*) and let $L_k$ be the set of frequent patterns containing $k$ items (i.e., *frequent $k$-itemsets*). Note that $L_k \subseteq C_k$, since all frequent patterns are candidate patterns but the reverse is not necessarily true. The first step in the Apriori algorithm is to determine $L_1$ (i.e., frequent 1-itemsets). This is accomplished by scanning through each transaction and counting the number of occurrences of each item in the transaction database. The frequent singletons are then discovered to be the domain items having at least *minsup* occurrences.

For the remainder of the Apriori algorithm, it repeatedly uses frequent $(k-1)$-itemsets to generate candidate $k$-itemsets where $k$ is an integer with $k \geq 2$ (i.e., $2 \leq k \in \mathbb{Z}$), and then discovers which of those candidate patterns are frequent. In the main loop of the Apriori algorithm, we start by setting $k = 2$. It initially computes $C_k$ from $L_{k-1}$ by performing a self-join on $L_{k-1}$. If the first $k-2$ items in two frequent $(k-1)$-itemsets in $L_{k-1}$ are the same, then it generates a candidate $k$-itemset in $C_k$ containing those $k-2$ items and the last item of those 2 itemsets. After initially creating $C_k$, it prunes $C_k$ by removing from $C_k$ any candidate $k$-itemsets having at least one $(k-1)$-subset not belonging to $L_{k-1}$.

Example 1. *If $L_2 = \{\{a, b\}, \{a, c\}\}$, then Apriori generates $\{a, b, c\} \in C_3$ by joining $\{a, b\}$ and $\{a, c\}$. However, it prunes $\{a, b, c\}$ from $C_3$ because a subset $\{b, c\} \notin L_2$.*

Next, Apriori counts the support (i.e., number of occurrences) of each candidate $k$-itemset in $C_k$ by scanning through each transaction and determining which candidate $k$-itemsets occur in that transaction. It is of interest to note that there are ways to speed up the counting process for the Apriori algorithm [64]. Afterwards, let $L_k$ be the set of itemsets in $C_k$ with a support at least *minsup*. Then, it then increments $k$ and repeats the previous steps. The loop terminates when no further patterns can be discovered for $L_{k-1}$. Consequently, it returns $\bigcup_k L_k$ (i.e., union of all $L_k$) as all the frequent patterns.

## 2.2 Vertical Boolean Frequent Pattern Mining with the Eclat Algorithm

Recall from Section 1 that the Eclat (Equivalence CLAss Transformation) algorithm [61] is an example of vertical item-centric frequent pattern mining algorithms, with which data are represented as a collection of transaction ID sets (i.e., *tidsets*). Each tidset for an item captures which transactions contain the specific item. The presence of the transaction ID in the $tidset(X)$ indicates the corresponding transaction contains the item $X$, whereas the absence of the transaction ID from the $tidset(X)$ indicates the corresponding transaction does not contain the item $X$.

Let us discuss the difference between the horizontal transaction database and the vertical transaction database. Horizontal transaction databases refer to the usual representation of transactions, where a set of items is associated with each transaction [63, 64]. The Apriori algorithm uses the horizontal representation. On the other hand, one can represent the transaction database in a "vertical" format [61]. A tidset for an item can represent a transaction database in a vertical format by adding a transaction ID to the tidset for indicating the presence of the item in the corresponding transaction.

Example 2. *For transactions $t_1 = \{a, b\}$ and $t_2 = \{b\}$ in a horizontal transaction database, the corresponding vertical representation of the transaction database is $\{t_1\} \subseteq tidset(\{a\})$ and $\{t_1, t_2\} \subseteq tidset(\{b\})$.*

The Eclat algorithm makes use of the vertical representation to mine frequent patterns. Like the Apriori algorithm, let $C_k$ and $L_k$ be the sets containing candidate and frequent $k$-itemsets, respectively. First, the Eclat algorithm discovers which itemsets are in $L_1$. It then computes the support of any candidate 1-itemset by counting the number of transaction IDs in its corresponding tidset. Mathematically, for a singleton $\{x\}$, $sup(\{x\}) = |tidset(\{x\})|$. After computing the support for every item occurring in the transaction database, $L_1$ contains singletons with a support $\geq$ *minsup*.

After discovering $L_1$, the main loop of the Eclat algorithm is be executed in a similar fashion as in the Apriori algorithm. Consider the first loop iteration with $k = 2$. The first part of the loop involves generating $C_k$ from $L_{k-1}$ by using the same candidate generation method as in the Apriori algorithm (i.e., performing a self-join on $L_{k-1}$ and pruning the resulting set). Next, it forms a tidset corresponding to each candidate $k$-itemset in $C_k$. Suppose that, for some candidate $k$-itemset $X \in C_k$, $W$ is a $(k-2)$-itemset containing the first $k-2$ items in $X$, $y$ is the second last item in $X$, and $z$ is the last item in $X$. Then, $X = W \cup \{y\} \cup \{z\}$. The algorithm computes the tidset of $X$ as the set intersection of $(W \cup \{y\})$ and $(W \cup \{z\})$, i.e., $tidset(X) = tidset(W \cup \{y\}) \cap tidset(W \cup \{z\})$. Next, it computes the support of each pattern in $C_k$ by counting the number of transaction IDs in the resulting set intersection, i.e., $sup(X) = |tidset(X)|$. The frequent $k$-itemsets in $L_k$ are computed as the candidate $k$-itemsets in $C_k$ having a support $\geq$ *minsup*. At the end of a loop iteration, increase $k$ by 1 and continue iterating through the main loop (if necessary). The loop stops iterating when $L_{k-1}$ is empty. In a similar fashion to the Apriori algorithm, $\bigcup_k L_k$ is returned as frequent patterns.

## 2.3 Quantitative Association Rule Mining

For quantitative association rule mining [65, 66], suppose that $I = \{i_1, i_2, \ldots, i_m\}$ is the set of all items that can be found in a transaction database for some positive integer $m \in \mathbb{Z}^+$. Then, a transaction can be represented as $t = \{(e_1, f_1), (e_2, f_2), \ldots, (e_s, f_s)\}$ for some $s \in \mathbb{Z}^+$ where

- each item $e_i \in I$ such that $e_i \neq e_j$ whenever $i \neq j$, and
- each quantity $f_i \in \mathbb{Z}^+$.

The quantitative transaction database is $D = (t_1, t_2, \ldots, t_n)$, which is the set of all transactions. Each transaction has a unique transaction ID. An item-expression—or **itemexp** for short—is an ordered triplet of the form $(p\theta q)$, where $p \in I, \theta \in \{=, \geq, \leq\}$, and $q \in \mathbb{Z}^+$. Then, a set of item expressions—or **itemexpset** for short—can be defined as a set $X = x_1, x_2, \ldots, x_k$ for some $k \in \mathbb{Z}^+$ where each $x_i = (p_i \theta_i q_i)$ is an itemexp such that $p_i \neq p_j$ whenever $i \neq j$. Then, $t$ satisfies $X$ if $\forall i \in \{1, 2, \ldots, k\}, \exists j \in \{1, 2, \ldots, s\}$ such that $p_i = e_j$ and the expression $(f_j \theta_i q_i)$ is true. If an itemexpset $X$ contains an itemexp of the form $(p \leq q)$ where $p \in I$ and $q \in \mathbb{Z}^+$, then for a transaction $t$ to satisfy $X$, item $p$ must still occur in $t$ at least once, even though $0 < q$. In other words, the number of occurrences of item $p \in t$ must be in the interval $[1, q]$. By including this restriction, many itemexpsets are prevented from being considered where an item can occur zero times.

Example 3. *For a transaction $t_1 = \{(a, 2), (b, 3), (c, 1)\}$ (which captures 2 occurrences of item a, 3 occurrences of item b, and 1 occurrence of item c), it satisfies the itemexpset $X_1 = \{(a = 2), (b \geq 1)\}$. However, $t_1$ does not satisfy $X_2 = \{(a \leq 2), (c \geq 2)\}$ because $X_2$ requires the quantity of c at least 2 but c only occurs once in $t_1$ (i.e., quantity of $c = 1$). Similarly, $t_2 = \{(a, 1)\}$ also does not satisfy $X_3 = \{(b \leq 2)\}$ because $X_3$ requires $0 <$ quantity of $b \leq 2$ but b does not occur in $t_2$ (i.e., quantity of $b = 0$).*

The support $sup(X)$ of an itemexpset $X$ is defined to be the number of transactions (in $D$) satisfying $X$. Now, let *minsup* be some non-negative real number, i.e., $minsup \in \mathbb{R}^+ \cup \{0\}$. Then, $X$ is a frequent itemexpset if $sup(X) \geq minsup$.

As association rules can be defined for Boolean frequent pattern mining, they can also be defined for quantitative frequent pattern mining. For two itemexpsets $X$ and $Y$, the association rule $X \rightarrow$Y is interesting if:

- there are no common items between $X$ and $Y$,
- $sup(X \rightarrow Y) = sup(X \cup Y) \geq minsup$, and
- $conf(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \geq minconf \in [0, 1]$,

Example 4. *Association rule $\{(a \geq 2)\} \rightarrow \{(b \leq 3), (c = 1)\}$ can be interesting if its support and confidence values can be satisfied. However, a rule $\{(a = 5)\} \rightarrow \{(a \geq 3)\}$, which reveals that a customer who purchases exactly 5 orders of item a is likely to purchase at least 3 orders of a, cannot be interesting because item a is common on both sides of the rule.*

## 2.4 Horizontal Quantitative Frequent Pattern Mining with the MQA-M Algorithm

As an algorithm for mining quantitative frequent patterns, the MQA-M (Mining Quantitative Association rules with Multiple comparison operators) [65] is similar to the Apriori algorithm except that it is generalized to handle quantitative transaction databases. For any $k \in \mathbb{Z}^+$, let $C_k$ be the set of candidate itemexpsets containing $k$ itemexps, and let $L_k$ be the set of frequent itemexpsets containing $k$ itemexps. Like in the Apriori algorithm, $L_k \subseteq C_k$. The MQA-M algorithm starts by generating $C_1$. Suppose that $itemmax[p]$ represents the maximum number of times an item $p$ appears in a transaction.

Example 5. *If a quantitative database consists of two transactions* $t_1 = \{(a, 1)\}$ *and* $t_2 = \{(a, 3)\}$, *then* $itemmax[a] = 3$ *because the highest number of times a appears in a transaction is 3.*

Afterwards, for each item $p$ appearing in the quantitative transaction database, add every itemexpset of the form $\{(p, \theta, q)\}$ to $C_1$, where $\theta \in \{=, \geq, \leq\}$ and $q \in \{1, \ldots, itemmax[p]\}$. The algorithm computes the support of each itemexpset in $C_1$ by iterating through the transactions and incrementing the support of an itemexpset in $C_1$ if the transaction satisfies the itemexpset. Let $k = 1$. Then, $L_1$ becomes the set of all itemexpsets with a support $\geq$ *minsup*. The algorithm removes some itemexpsets from $L_1$ by using two pruning rules [65]:

1. Suppose that $X$ contains an itemexp of the form $(z \leq r)$, where $z$ is an item and $r \in \mathbb{Z}^+$. This first pruning rule states that, if there is another itemexpset $Y \in L_k$ with the same support as $X$ except that $(z \leq r)$ is replaced by $(z \leq r + 1)$, then $Y$ can be pruned from $L_k$.
2. Suppose that $X$ contains an itemexp of the form $(z \geq r)$, where $z$ is an item and $r \in \mathbb{Z}^+$. This second pruning rule states that if there is another itemexpset $Y \in L_k$ with the same support as $X$ except that $(z \geq r)$ is replaced by $(z \geq r - 1)$, then $Y$ can be pruned from $L_k$.

Like the Apriori algorithm, the MQA-M also has a main loop. It first runs the loop with $k = 2$. The loop body begins with generating $C_k$ from $L_{k-1}$. $C_k$ is initially generated using a self-join on $L_{k-1}$. If two itemexpsets in $L_{k-1}$ have the same first $(k - 2)$ itemexps, then it generates an itemexpset in $C_k$ consisting of those (k-2) itemexps and the last itemexp in the two itemexpsets in $L_{k-1}$. However, it imposes an additional restriction that it does not create an itemexpset in $C_k$ where there are two itemexps referring to the same item. After the join step, it prunes itemexpsets from $C_k$ with a subset containing $(k - 1)$ itemexps where that subset is not in $L_{k-1}$. It gets $L_k$ from $C_k$ using the same procedure that was used to obtain $L_1$. It uses the two aforementioned pruning rules to removes some itemsets from $L_k$. At the end of the loop body, it increments $k$ and repeats the previous steps (if necessary). The loop terminates when $L_{k-1}$ is empty. Afterwards, it returns $\bigcup_k L_k$, which contains all the interesting frequent itemexpsets.

Example 6. *Although* $L_1$ *contains* $\{(a = 1)\}$ *and* $\{(a \geq 2)\}$, *MQA-M does not form* $\{(a = 1), (a \geq 2)\} \in C_2$.

# 3 VERTICAL QUANTITATIVE FREQUENT PATTERN MINING WITH OUR Q-ECLAT ALGORITHM

## 3.1 Vertical Representation of Quantitative Data

To represent quantitative transaction databases in a vertical format, for each item that occurs in the transaction database, we store it as a set of pairs. Each pair contains a transaction ID associated with that item and the number of occurrences of the item in the transaction. Since we are storing a pair, we can call these sets "pairsets". It is useful to convert the quantitative transaction database to this vertical format when implementing the Q-Eclat algorithm.

Example 7. *A horizontal database containing two transactions* $t_1 = \{(a, 1)\}$ *and* $t_2 = \{(a, 3)\}$ *can be represented vertically using* $pairset(a) = \{(t_1, 1), (t_2, 3)\}$. *Then,*

For quantitative association rule mining, we define $tidset(X)$ of any itemexpset $X$ to be the set of transaction IDs corresponding to transactions which satisfy $X$. When $X$ is an itemexpset containing at least two itemexps, we can break down $X = W \cup \{y\} \cup \{z\}$ where (a) $W$ is an itemexpset with two fewer elements than $X$ and (b) $y$ and $z$ are itemexps. Like tidsets for Boolean frequent itemset mining, we have the recursive equation: $tidset(X) = tidset(W \cup \{y\}) \cap tidset(W \cup \{z\})$. We will use this equation to generate tidsets for itemexpsets containing at least two elements when running our Q-Eclat algorithm. The support of an itemexpset X can be computed by counting the number of elements in its tidset, i.e., $sup(X) = |tidset(X)|$.

## 3.2 Q-Eclat Algorithm

Here, let us describe how our Q-Eclat algorithm discovers quantitative frequent patterns vertically. For any integer $k \geq 1$, define $C_k$ to be the set of candidate $k$-itemexpsets and $L_k$ to be the set of frequent $k$-itemexpsets. First, we convert the quantitative transaction database into a vertical format if it is in its horizontal format. The vertical format is useful for computing the tidsets corresponding to the candidate 1-itemexpsets (i.e., $C_1$). The next step of our algorithm is to compute all candidate 1-itemexpsets in $C_1$. Each of those itemexpsets consists of a single itemexp of the form

$$(item, operation, quantity)$$

where

- *item* is an item in the transaction database,
- *operator* $\theta \in \{=, \geq, \leq\}$, and
- *quantity* $q \in \{1, \ldots, itemmax[item]\}$.

We compute $itemmax[item]$ as the maximum number of times an item appears in a transaction, over all transactions in the transaction database.

After computing $C_1$, we compute the tidsets associated with each candidate 1-itemexpset. The tidsets can be computed from the vertical representation of the quantitative transaction database. We then compute the support of each candidate 1-itemexpset by counting the elements in its corresponding tidset. The frequent 1-itemexpsets are candidate 1-itemexpsets having a support $\geq$ *minsup*.

Finally, we remove some itemexpsets from $L_1$ based on our two *new pruning rules*, which will be described in Section 3.3.

Then, we set $k = 2$ and begin executing the main loop. The first step in the main loop body is to generate $C_k$ using $L_{k-1}$. We initially create $C_k$ by performing a self-join on $L_{k-1}$. If there are two frequent $(k-1)$-itemexpsets in $L_{k-1}$ where their first $(k-2)$-itemexps are the same and their last itemexp refer to different items, then we add to $C_k$ a candidate $k$-itemexpset that consists of the first $(k-2)$-itemexps and the last itemexp of both itemexpsets. Afterwards, we prune any candidate $k$-itemexpset from $C_k$ that contains a sub-itemexpset with $(k-1)$-itemexps that do not belong to $L_{k-1}$. The next step is to create tidsets corresponding to every candidate $k$-itemexpset in $C_k$. This can be done using the recursive definition for tidsets:

$$tidset(X) = tidset(W \cup \{y\}) \cap tidset(W \cup \{z\}) \qquad (1)$$

After computing the tidsets, we compute the support of each candidate $k$-itemexpset in $C_k$. Any candidate $k$-itemexpset having a support $\geq$ *minsup* is added to $L_k$. Using the two pruning rules, we remove some uninteresting itemexpsets from $L_k$ if necessary. After pruning $L_k$, we have reached the end of the loop body. Hence, we increment $k$ and repeat the main steps again if necessary. The main loop stops running once $L_k$ is empty. Our Q-Eclat algorithm returns $\bigcup_k L_k$, which contains all interesting frequent itemexpsets.

## 3.3 Our New Pruning Rules for Q-Eclat Algorithm

As mentioned in Section 2.4, there were two pruning rules (for the MQA-M algorithm) to remove unnecessary itemexpsets from $L_k$ where integer $k \geq 1$. Here, we present two new pruning rules that are more general than the original pruning rule. Our pruning rules remove some uninteresting itemexpsets, which were not removed in the original pruning rules. Let $X$ be an itemexpset in $L_k$. Then, our pruning rules are described as follows:

1. Suppose that $X$ contains an itemexp of the form $(z \leq r)$, where $z$ is an item and $r \in \mathbb{Z}^+$. Our first pruning rule states that, if there is another itemexpset $Y \in L_k$ with the same support as $X$ except that $(z \leq r)$ is replaced by $(z \leq r + s)$ for some $s \in \mathbb{Z}^+$, then $Y$ can be pruned from $L_k$.
2. Suppose that $X$ contains an itemexp of the form $(z \geq r)$, where $z$ is an item and $r \in \mathbb{Z}^+$. Our second pruning rule states that if there is another itemexpset $Y \in L_k$ with the same support as $X$ except that $(z \geq r)$ is replaced by $(z \geq r - s)$ for some $s \in \mathbb{Z}^+$, then $Y$ can be pruned from $L_k$.

A key difference between the original pruning rules and our new pruning rules is that the new pruning rules can handle differences in quantity greater than 1. Instead of considering itemexpsets of the form $(z \leq r + 1)$ or $(z \geq r - 1)$, we consider the more general cases of $(z \leq r + s)$ or $(z \geq r - s)$ for some positive integer $s$. As a result, these rules eliminate at least as many itemexpsets from $L_k$ as the original pruning rules. Observed from Example 8, our improved pruning rules are more powerful in removing redundant frequent itemexpsets.

Example 8. *Suppose that $L_2$ contains two frequent 2-itemexpsets $\{(a \geq 2), (b = 1)\}$ and $\{(a \geq 4), (b = 1)\}$ before pruning and they have the same support. Using the original pruning rules used in MQA-M, neither itemexpset would be pruned. In contrast, when using our new pruning rules, $\{(a \geq 2), (b = 1)\}$ would be pruned from $L_2$.*

Example 9. *Suppose we set minsup=1 and we have three transactions in a quantitative transaction database:*

- $t_1 = \{a : 2\}$,
- $t_2 = \{a : 4, b : 1\}$, *and*
- $t_3 = \{b : 1\}$.

*Then, we observe the occurrences of each domain item in the transaction and compute its itemmax:*

- $itemmax[a] = 4$ *because the highest number of occurrences of a in a transaction is 4 (in transaction $t_2$), and*
- $itemmax[b] = 1$ *because the highest number of occurrences of b in a transaction is 1 (in both transactions $t_2$ and $t_3$).*

*To generate $C_1$, we must generate every combination of an item, comparison operation, and quantity. There are two items (i.e., a and b) and three operators (i.e., $=, \geq$ and $\leq$). For item a, there are four quantity values (i.e., from 1 to $itemmax[a] = 4$). This leads to a total of $1 \times 3 \times 4 = 12$ candidate itemexpsets from item a. Similarly, the one quantity value (due to $itemmax[b] = 1$) leads to a total of $1 \times 3 \times 1 = 3$ candidate itemexpsets from item b. Consequently, this leads to a total of $12 + 3 = 15$ candidate itemexpsets as shown in the first column of Table 1.*

*For each itemexpset in $C_1$, we compute its corresponding tidsets. The support of those itemexpsets is equal to the number of elements (i.e., transaction IDs) in the tidset. We present the itemexpsets X in $C_1$, their tidsets, and their supports in Table 1.*

**Table 1: Candidate and frequent 1-itemexpsets**

| $X \in C_1$ | $tidset(X)$ | $sup(X)$ | $\geq$ *minsup* | *interesting* |
|---|---|---|---|---|
| $\{(a = 1)\}$ | $\emptyset$ | 0 | | |
| $\{(a = 2)\}$ | $\{t_1\}$ | 1 | √ | √ |
| $\{(a = 3)\}$ | $\emptyset$ | 0 | | |
| $\{(a = 4)\}$ | $\{t_2\}$ | 1 | √ | √ |
| $\{(a \geq 1)\}$ | $\{t_1, t_2\}$ | 2 | √ | |
| $\{(a \geq 2)\}$ | $\{t_1, t_2\}$ | 2 | √ | √ |
| $\{(a \geq 3)\}$ | $\{t_2\}$ | 1 | √ | |
| $\{(a \geq 4)\}$ | $\{t_2\}$ | 1 | √ | √ |
| $\{(a \leq 1)\}$ | $\emptyset$ | 0 | | |
| $\{(a \leq 2)\}$ | $\{t_1\}$ | 1 | √ | √ |
| $\{(a \leq 3)\}$ | $\{t_1\}$ | 1 | √ | |
| $\{(a \leq 4)\}$ | $\{t_1, t_2\}$ | 2 | √ | √ |
| $\{(b = 1)\}$ | $\{t_2, t_3\}$ | 2 | √ | √ |
| $\{(b \geq 1)\}$ | $\{t_2, t_3\}$ | 2 | √ | √ |
| $\{(b \leq 1)\}$ | $\{t_2, t_3\}$ | 2 | √ | √ |

*Among these 15 candidate 1-itemexpsets, only 12 of them satisfy minsup = 1. We obtain initial $L_1$ by only keeping these 12 candidate 1-itemexpsets having support $\geq$ minsup. They are listed on the fourth column of Table 1.*

*Then, by using our pruning rules described in Section 3.3, we further prune away three more redundant 1-itemexpsets:*

- *Prune 1-itemexpset $\{(a \geq 1)\}$ by Pruning Rule (2) because both $\{(a \geq 1)\} \in L_1$ and $\{(a \geq 2)\} \in L_1$ have the same support;*

**Table 2: Candidate and frequent 2-itemexpsets**

| $X \in C_2$ | *tidset*($X$) | *sup*($X$) | ≥ *minsup* | *interesting* |
|---|---|---|---|---|
| $\{(a=2),(b=1)\}$ | $\emptyset$ | *0* | | |
| $\{(a=2),(b\geq 1)\}$ | $\emptyset$ | *0* | | |
| $\{(a=2),(b\leq 1)\}$ | $\emptyset$ | *0* | | |
| $\{(a=4),(b=1)\}$ | $\{t_2\}$ | *1* | √ | √ |
| $\{(a=4),(b\geq 1)\}$ | $\{t_2\}$ | *1* | √ | √ |
| $\{(a=4),(b\leq 1)\}$ | $\{t_2\}$ | *1* | √ | √ |
| $\{(a\geq 2),(b=1)\}$ | $\{t_2\}$ | *1* | √ | |
| $\{(a\geq 2),(b\geq 1)\}$ | $\{t_2\}$ | *1* | √ | |
| $\{(a\geq 2),(b\leq 1)\}$ | $\{t_2\}$ | *1* | √ | |
| $\{(a\geq 4),(b=1)\}$ | $\{t_2\}$ | *1* | √ | √ |
| $\{(a\geq 4),(b\geq 1)\}$ | $\{t_2\}$ | *1* | √ | √ |
| $\{(a\geq 4),(b\leq 1)\}$ | $\{t_2\}$ | *1* | √ | √ |
| $\{(a\leq 2),(b=1)\}$ | $\emptyset$ | *0* | | |
| $\{(a\leq 2),(b\geq 1)\}$ | $\emptyset$ | *0* | | |
| $\{(a\leq 2),(b\leq 1)\}$ | $\emptyset$ | *0* | | |
| $\{(a\leq 4),(b=1)\}$ | $\{t_2\}$ | *1* | √ | √ |
| $\{(a\leq 4),(b\geq 1)\}$ | $\{t_2\}$ | *1* | √ | √ |
| $\{(a\leq 4),(b\leq 1)\}$ | $\{t_2\}$ | *1* | √ | √ |

- *Prune 1-itemexpset* $\{(a\geq 3)\}$ *by Pruning Rule (2) because both* $\{(a\geq 3)\}\in L_1$ *and* $\{(a\geq 4)\}\in L_1$ *have the same support; and*
- *Prune 1-itemexpset* $\{(a\leq 3)\}$ *by Pruning Rule (1) because both* $\{(a\leq 2)\}\in L_1$ *and* $\{(a\leq 3)\}\in L_1$ *have the same support.*

*Hence, the final* $L_1$ *ends up with only* $12-3=9$ *frequent but not redundant 1-itemexpsets:*

- $\{(a=2)\}$,
- $\{(a=4)\}$,
- $\{(a\geq 2)\}$,
- $\{(a\geq 4)\}$,
- $\{(a\leq 2)\}$,
- $\{(a\leq 4)\}$,
- $\{(b=1)\}$,
- $\{(b\geq 1)\}$,
- $\{(b\leq 1)\}$.

*Afterwards, the main loop is executed with* $k=2$. *We begin with the generation of* $C_2$. *The first step in generating* $C_2$ *is to perform a self-join on* $L_1$. *In this scenario, this means getting pairs of itemexps where the itemexps refer to different items. This yields* $6\times 3=18$ *different candidate 2-itemexpsets in* $C_2$ *as shown in Table 2.*

*Among these 18 candidate 2-itemexpsets, only 12 of them satisfy* $minsup=1$. *We obtain initial* $L_2$ *by only keeping these 12 candidate 2-itemexpsets having support* ≥ *minsup. They are listed on the fourth column of Table 2.*

*Note that none of these 12 itemexpsets in the initial* $L_2$ *can be pruned by the* original *pruning rules used in the MQA-M algorithm. For instance, recall from Example 8, although both* $\{(a\geq 2),(b=1)\}\in L_2$ *and* $\{(a\geq 4),(b=1)\}\in L_2$ *have the same support,* $\{(a\geq 2),(b=1)\}$ *would not be pruned. However, by using our pruning rules described in Section 3.3, we can further prune away three more redundant 2-itemexpsets:*

- *Prune 2-itemexpset* $\{(a\geq 2),(b=1)\}$ *by our new Pruning Rule (2) because both* $\{(a\geq 2),(b=1)\}\in L_2$ *and* $\{(a\geq 4),(b=1)\}\in L_2$ *have the same support;*
- *Prune 1-itemexpset* $\{(a\geq 2),(b\geq 1)\}$ *by our new Pruning Rule (2) because both* $\{(a\geq 2),(b\geq 1)\}\in L_2$ *and* $\{(a\geq 4),(b\geq 1)\}\in L_2$ *have the same support; and*
- *Prune 1-itemexpset* $\{(a\geq 2),(b\leq 1)\}$ *by our new Pruning Rule (2) because both* $\{(a\geq 2),(b\leq 1)\}\in L_2$ *and* $\{(a\geq 4),(b\leq 1)\}\in L_2$ *have the same support.*

*Hence, the final* $L_2$ *ends up with only* $12-3=9$ *frequent but not redundant 2-itemexpsets:*

- $\{(a=4),(b=1)\}$,
- $\{(a=4),(b\geq 1)\}$,
- $\{(a=4),(b\leq 1)\}$,
- $\{(a\geq 4),(b=1)\}$,
- $\{(a\geq 4),(b\geq 1)\}$,
- $\{(a\geq 4),(b\leq 1)\}$,
- $\{(a\leq 4),(b=1)\}$,
- $\{(a\leq 4),(b\geq 1)\}$,
- $\{(a\leq 4),(b\leq 1)\}$.

*With only two items* $a$ *and* $b$, *no candidate 3-itemexpsets can be formed. Consequently, our Q-Eclat returns the nine frequent but not redundant 1-itemexpsets and the nine other frequent but not redundant 2-itemexpsets, for a total of 18 itemexpsets as the output.*

## 4 EVALUATION

To evaluate our Q-Eclat algorithm, we compared it with the existing MQA-M algorithm [65]. The performance of the algorithms is assessed using four different quantitative transaction databases:

- two synthetic datasets: Here, we assume that there are $n$ transactions and $|I|$ different items. Each item has a probability prob of occurring in a particular transaction, where $0\leq prob\leq 1$. If the item appears in the transaction, then

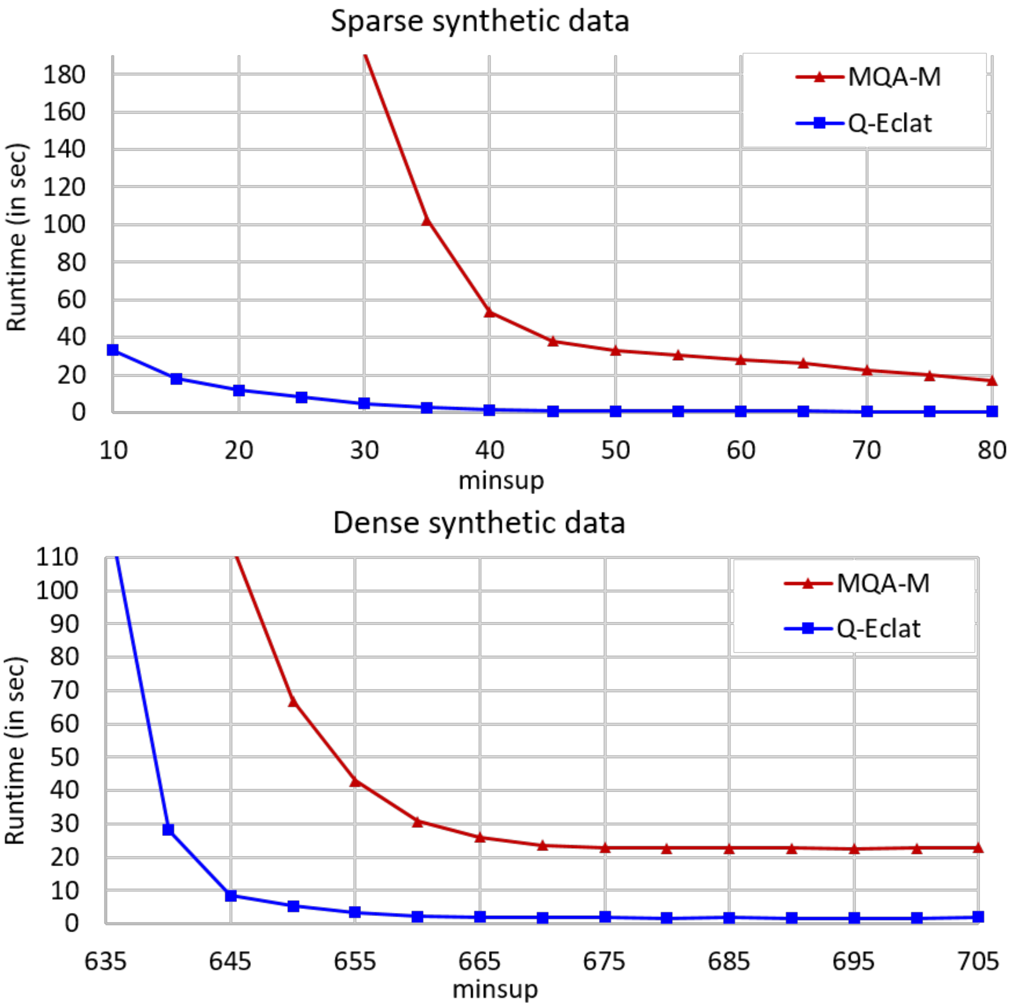


**Figure 1: Runtimes of the existing MQA-M algorithm and our Q-Eclat algorithm for quantitative frequent pattern mining with various *minsup* values on *synthetic* datasets: (a) sparse and (b) dense synthetic datasets.**

the number of occurrences of that item follows a Poisson($\lambda$) distribution plus 1. We set $n = 1000$, $|I| = 50$, and $\lambda = 1$. The values of *prob* for these two quantitative transaction databases are 0.2 and 0.8. These quantitative transaction databases considered as sparse and dense, respectively.

- two real-life datasets from UCI ML Repository [67]: Here, we modified the chess and mushroom datasets to make them quantitative transaction databases. Whenever an item occurs in a transaction, instead of it only occurring once, its number of occurrences follows a Poisson($\lambda$) distribution plus 1.

More specifically, these two synthetic datasets and two real-life datasets are:

(1) sparse synthetic dataset, with *prob* = 0.2;
(2) dense synthetic dataset, with *prob* = 0.8;
(3) modified real-life chess dataset; and
(4) modified real-life mushroom dataset.

The two algorithms for quantitative frequent itemset mining (i.e., MQA-M [65] and our Q-Eclat) have been implemented in the Python language. The algorithms were run on a Windows 10 Nitro AN515-55 laptop using an Intel Core i5-10300H CPU at 2.50 GHz and 8.00 GB RAM. To keep the comparisons between the algorithms fair, we used many of the same functions between the algorithms. These include generation of candidate itemexpsets, discovery of frequent itemexpsets, and application of our pruning rules on the frequent itemexpsets. When we implement the MQA-M algorithm, we use our improved pruning rules used in Q-Eclat rather than the pruning rules originally used with MQA-M. This allows the simulations to emphasize the differences between the algorithms.

We run the main code for each of the aforementioned quantitative transaction datasets. For each quantitative transaction database, we use a sequence of *minsup* values. The sequence depends on the quantitative transaction database being used to observe interesting results that the algorithms did not take too long to run. For each

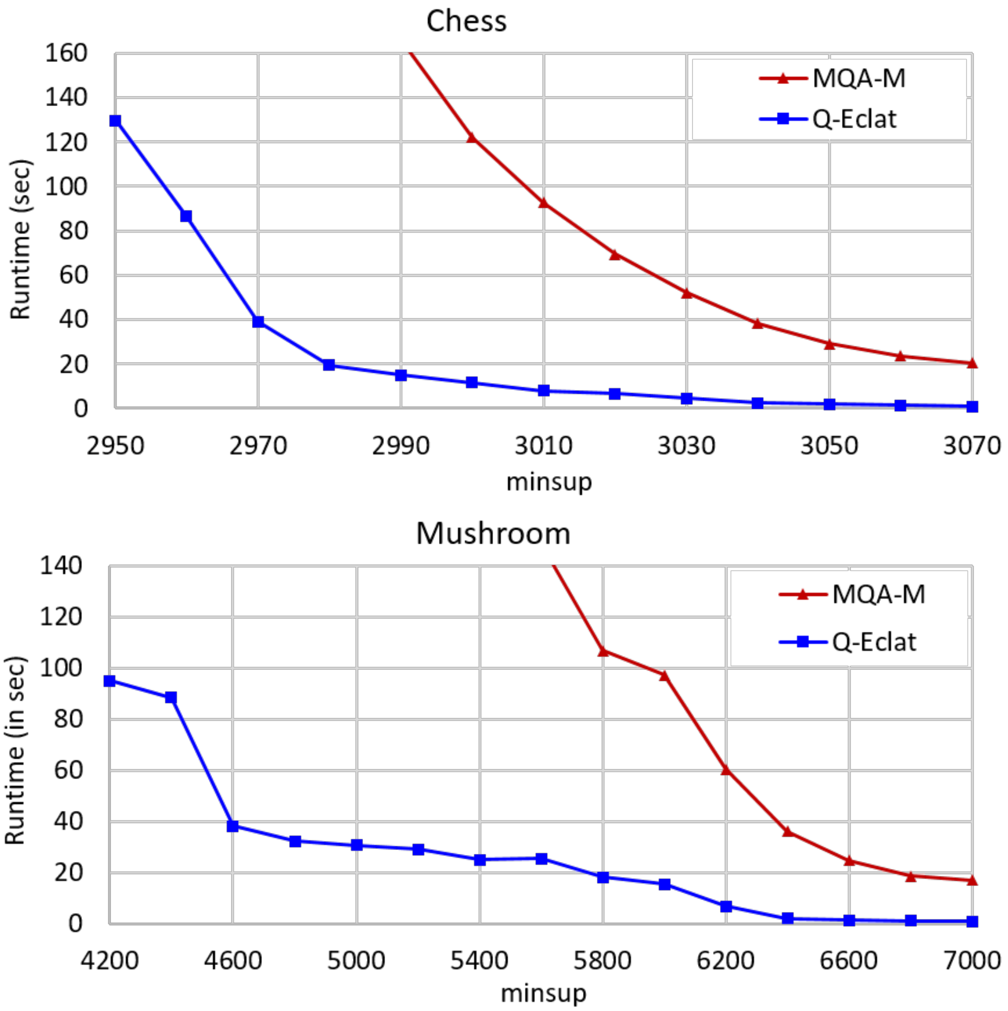


**Figure 2: Runtimes of the existing MQA-M algorithm and our Q-Eclat algorithm for quantitative frequent pattern mining with various *minsup* values on *real-life* datasets: (a) chess and (c) mushroom datasets.**

combination of a quantitative transaction database and a value for *minsup*, the MQA-M and Q-Eclat algorithms were run and timed. Reported runtimes were average of multiple runs.

Figs. 1 and 2 show the runtimes of each of the two algorithms for a variety of values of *minsup* for each of the four quantitative transaction datasets. The runtime (in seconds) is shown on the $y$-axis while the value of *minsup* is given on the $x$-axis. In all cases, our Q-Eclat took shorter runtimes than the existing MQA-M algorithm to return the same collections of itemsexpsets.

## 5 CONCLUSIONS

In this paper, we presented our vertical quantitative frequent itemset mining called Q-Eclat. This Q-Eclat algorithm first represents the big data as a collection of sets of transaction IDs (i.e., tidsets). Each item-centric tidset captures the IDs of transactions containing the item, as well as the quantity of that item in each transaction. With this representation, our algorithm then vertically mines quantitative frequent patterns. During the mining process, our new pruning rules reduce the mining space, and thus shorten the runtime. When compared the existing MQA-M algorithm (which was built for quantitative frequent pattern mining), evaluation results show that our quantitative vertical Q-Eclat algorithm takes shorter runtime to mine quantitative frequent patterns. As *ongoing and future work*, we explore ways to further enhance the mining of quantitative frequent patterns and to extend this work to mine other quantitative patterns.

## ACKNOWLEDGMENTS

This work is partially supported by the (a) Natural Sciences and Engineering Research Council of Canada (NSERC), as well as the (b) University of Manitoba. Thanks Junyi Andre Lu, Denys Popov, and Sanskar Raval—all from University of Manitoba—for their assistance in parts of this work.

# REFERENCES

[1] S. Sahri, R. Moussa. 2021. Customized eager-lazy data cleansing for satisfactory big data veracity. In IDEAS 2021, pp. 157-165.

[2] Y. Zhao, et al. 2021. A zone-based data lake architecture for IoT, small and big data. In IDEAS 2021, pp. 94-102.

[3] M. Asiri, et al. 2018. Feature reduction improves classification accuracy in healthcare. In IDEAS 2018, pp. 193-198.

[4] C.K. Leung, et al. 2020. Data science for healthcare predictive analytics. In IDEAS 2020, pp. 8:1-8:10.

[5] C.K. Leung, et al. 2022. Deep learning based multi-label prediction of hospitalization for COVID-19 cases. In IEEE CBMS 2022, pp. 96-101.

[6] C.K. Leung, et al. 2022. Towards trustworthy artificial intelligence in healthcare. In IEEE ICHI 2022, pp. 626-632.

[7] N.D.T. Tran, et al. 2022. A deep learning based predictive model for healthcare analytics. In IEEE ICHI 2022, pp. 547-549.

[8] D.L.X. Fung, et al. 2021. Self-supervised deep learning model for COVID-19 lung CT image segmentation highlighting putative causal relationship among age, underlying disease and COVID-19. BMC Journal of Translational Medicine 19, pp. 318:1-318:18.

[9] C.K. Leung, C. Zhao. 2021. Big data intelligence solution for health analytics of COVID-19 data with spatial hierarchy. In IEEE DataCom 2021, pp. 13-20.

[10] C.K. Leung, et al. 2021. Smart data analytics on COVID-19 data. In IEEE iThings-GreenCom-CPSCom-SmartData-Cybermatics 2021, pp. 372-379.

[11] BA. Monchka, et al. 2022. The effect of disease co-occurrence measurement on multimorbidity networks: a population-based study. BMC Medical Research Methodology 22, pp. 165:1-165:16.

[12] J. Souza, et al. 2020. An innovative big data predictive analytics framework over hybrid big data sources with an application for disease analytics. In AINA 2020. AISC, vol. 1151, pp. 669-680.

[13] A.A. Audu, et al. 2019. An intelligent predictive analytics system for transportation analytics on open data towards the development of a smart city. In CISIS 2019. AISC, vol. 993, pp. 224-236.

[14] M.K. Mufida, et al. 2021. Towards a continuous forecasting mechanism of parking occupancy in urban environments. In IDEAS 2021, pp. 263-272.

[15] B. Nguyen, et al. 2022. A data science solution for mining weather data and transportation data for smart cities. In IEEE COMPSAC 2022, pp. 1672-1677.

[16] S. Ahn, et al. 2019. A fuzzy logic based machine learning tool for supporting big data business analytics in complex artificial intelligence environments. In FUZZ-IEEE 2019, pp. 1259-1264.

[17] D. Choudhery, C.K. Leung. 2014. Social media mining: prediction of box office revenue. In IDEAS 2017, pp. 20-29.

[18] C.K. Leung, et al. 2014. A machine learning approach for stock price prediction. In IDEAS 2014, pp. 274-277.

[19] K.J. Morris, et al. 2018. Token-based adaptive time-series prediction by ensembling linear and non-linear estimators: a machine learning approach for predictive analytics on big stock data. In IEEE ICMLA 2018, pp. 1486-1491.

[20] B. Vide, et al. 2021. Designing a business view of enterprise data: an approach based on a decentralised enterprise knowledge graph. In IDEAS 2021, pp. 184-193.

[21] C. Aarts, et al. 2020. A practical application for sentiment analysis on social media textual data. In IDEAS 2020, pp. 26:1-26:6.

[22] I. Afyouni, et al. 2020. Spatio-temporal event discovery in the big social data era. In IDEAS 2020, pp. 7:1-7:6.

[23] R.M. Cabusas, et al. 2022. Mining for fake news. In AINA 2022, Part II. LNNS, vol. 450, pp. 154-166.

[24] G. Bergami, et al. 2019. On approximate nesting of multiple social network graphs: a preliminary study. In IDEAS 2019, pp. 40:1-40:5.

[25] C.K. Leung. 2018. Mathematical model for propagation of influence in a social network. In Encyclopedia of Social Network Analysis and Mining, 2nd edn., pp. 1261-1269.

[26] C.K. Leung, et al. 2016. Parallel social network mining for interesting 'following' patterns. Concurrency and Computation: Practice & Experience 28(15), pp. 3994-4012.

[27] C.K. Leung, et al. 2018. Big data analytics of social network data: who cares most about you on Facebook? In Highlighting the Importance of Big Data Management and Analysis for Various Applications, pp. 1-15.

[28] R. Rouhi, et al. 2018. A cluster-based approach of smartphone camera fingerprint for user profiles resolution within social network. In IDEAS 2018, pp. 287-291.

[29] B.C. Isichei, et al. 2022. Sports data management, mining, and visualization. In AINA 2022, Part II. LNNS, vol. 450, pp. 141-153.

[30] C.S. Eom, et al. 2020. Effective privacy preserving data publishing by vectorization. Information Sciences 527, pp. 311-328.

[31] C.K. Leung. 2021. Data science for big data applications and services: data lake management, data analytics and visualization. In Big Data Analyses, Services, and Smart Data. AISC, vol. 899, pp. 28-44.

[32] R. Froese, et al. 2022. The border k-means clustering algorithm for one dimensional data. In IEEE BigComp 2022, pp. 35-42.

[33] J. Kim, et al. 2021. KNN-SC: novel spectral clustering algorithm using k-nearest neighbors. IEEE Access 9, pp. 152616-152627.

[34] B. Min, et al. 2020. Image classification for agricultural products using transfer learning. In BigDAS 2020, pp. 48-52.

[35] J.F. Smallwood, et al. 2022. Mining the impacts of COVID-19 pandemic on the labour market. In IMCOM 2022, 337-344.

[36] C.K. Leung, et al. 2021. Explainable data analytics for disease and healthcare informatics. In IDEAS 2021, pp. 65-74.

[37] S.P. Singh, et al. 2020. Analytics of similar-sounding names from the web with phonetic based clustering. In IEEE/WIC/ACM WI-IAT 2020, pp. 580-585.

[38] T. Fujimoto, et al. 2018. 3D visualization of data using SuperSQL and unity. In IDEAS 2018, pp. 141-147.

[39] C.S.H. Hoi, et al. 2022. Data, information and knowledge visualization for frequent patterns. In IV 2022, pp. 227-232.

[40] C.K. Leung, et al. 2011. Visual analytics of social networks: mining and visualizing co-authorship networks. In HCII-FAC 2011. LNCS (LNAI), vol. 6780, pp. 335-345.

[41] Y. Seong, et al. 2020. Guidelines for cybersecurity visualization design. In IDEAS 2020, pp. 25:1-25:6.

[42] M.T. Alam, et al. 2021. Discriminating frequent pattern based supervised graph embedding for classification. In PAKDD 2021, Part II. LNCS (LNAI), vol. 12713, pp. 16-28.

[43] M.T. Alam, et al. 2021. Mining frequent patterns from hypergraph databases. In PAKDD 2021, Part II. LNCS (LNAI), vol. 12713, pp. 3-15.

[44] C.K. Leung. 2019. Pattern mining for knowledge discovery. In IDEAS 2019, pp. 34:1-34:5.

[45] C.K. Leung, et al. 2012. A constrained frequent pattern mining system for handling aggregate constraints. In IDEAS 2012, pp. 14-23.

[46] M.T. Alam, et al. 2022. UGMINE: utility-based graph mining. Applied Intelligence. DOI: 10.1007/s10489-022-03385-8

[47] M.E.S. Chowdhury, et al. 2022. A new approach for mining correlated frequent subgraphs. ACM Transactions on Management Information Systems (TMIS) 13(1), pp. 9:1-9:28.

[48] K. Vaculik, L. Popelinsky. 2019. A genetic algorithm for discriminative graph pattern mining. In IDEAS 2019, pp. 46:1-46:2.

[49] S.D. Bernhard, et al. 2016. Clickstream prediction using sequential stream mining techniques with Markov chains. In IDEAS 2016, pp. 24-33.

[50] A. El Ouassouli, et al. 2019. Mining complex temporal dependencies from heterogeneous sensor data streams. In IDEAS 2019, pp. 23:1-23:10.

[51] M.S. Islam, et al. 2022. Discovering probabilistically weighted sequential patterns in uncertain databases. Applied Intelligence. DOI: 10.1007/s10489-022-03699-7

[52] C.K. Leung, et al. 2022. Visualization and visual knowledge discovery from big uncertain data. In IV 2022, pp. 336-341

[53] M.M Rahman, et al. 2019. Mining weighted frequent sequences in uncertain databases. Information Sciences 479, pp. 76-100.

[54] K.K. Roy, et al. 2021. Mining sequential patterns in uncertain databases using hierarchical index structure. In PAKDD 2021, Part II. LNCS (LNAI), vol. 12713, pp. 29-41.

[55] K.K. Roy, et al. 2022. Mining weighted sequential patterns in incremental uncertain databases. Information Sciences 582, pp. 865-896.

[56] S. Dawar, V. Goyal, 2015. UP-Hist tree: an efficient data structure for mining high utility patterns from transaction databases. In IDEAS 2015, pp. 56-61.

[57] P. Gupta, et al. 2021. Vertical data mining from relational data and its application to COVID-19 data. In Big Data Analyses, Services, and Smart Data. AISC, vol. 899, pp. 106-116.

[58] C.K. Leung, et al. 2012. Mining probabilistic datasets vertically. In IDEAS 2012, pp. 199-204.

[59] C.K. Leung, et al. 2018. Scalable vertical mining for big data analytics of frequent itemsets. In DEXA 2018, Part I. LNCS, vol. 11029, pp. 3-17.

[60] P. Shenoy, et al. 2000. Turbo-charging vertical mining of large databases. In ACM SIGMOD 2000, pp. 22-33.

[61] M.J. Zaki. 2000. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering 12(3), pp. 372-390.

[62] M.J. Zaki, K. Gouda. 2003. Fast vertical mining using diffsets. In ACM KDD 2003, pp. 326-335.

[63] R. Agrawal, et al. 1993. Mining association rules between sets of items in large databases. In ACM SIGMOD 1993, pp. 207-216.

[64] R. Agrawal, R. Srikant. 1994. Fast algorithms for mining association rules. In VLDB 1994, pp. 487-499.

[65] P.Y. Hsu, et al. 2004. Algorithms for mining association rules in bag databases. Information Sciences 166(1-4), pp. 31-47.

[66] R. Srikant, R. Agrawal. 1996. Mining quantitative association rules in large relational tables. In ACM SIGMOD 1996, pp. 1-12.

[67] D. Dua, C. Graff. 2019. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

# Identification of Weak Signals in a Temporal Graph of Social Interactions

Hiba Abou Jamra
LIB - EA 7534
Univ. Bourgogne Franche-Comté
Dijon, France
Hiba@depinfo.u-bourgogne.fr

Marinette Savonnet
LIB - EA 7534
Univ. Bourgogne Franche-Comté
Dijon, France
Marinette.Savonnet@u-bourgogne.fr

Éric Leclercq
LIB - EA 7534
Univ. Bourgogne Franche-Comté
Dijon, France
Eric.Leclercq@u-bourgogne.fr

## ABSTRACT

Social networks are becoming increasingly a source of wealth for people to connect with others in the society and express themselves. These networks store huge amounts of data related to individual and collective behavior, and relationships. Despite their importance, there exists few research that explains the factors leading to the evolution of these relationships, as well as abrupt changes in the behavior of individuals in contact. This paper proposes an approach based on the topology of social networks to detect early warnings of such changes, called weak signals. Our approach is in contrast to existing works that focus on analyzing major themes and trends, i.e. strong signals, prevalent in a social network at a particular point in time. We rely on a temporal interaction graph, and extract patterns that characterize weak signals. We demonstrate our approach and validate the detected signals through the analysis of social interactions between individuals of a captive Guinea baboons group, and confirm the existence of weak signals prior to the occurrence of an aggressive behavior.

## CCS CONCEPTS

• **Decision support systems** → **Data analytics**; • **Human-centered computing** → *Social networking sites*;

## KEYWORDS

Weak signals, social networks, topological analysis, graphlets

**ACM Reference Format:**
Hiba Abou Jamra, Marinette Savonnet, and Éric Leclercq. 2022. Identification of Weak Signals in a Temporal Graph of Social Interactions. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3548785.3548807

## 1 INTRODUCTION

The tremendous increase in diversity of available data provides perspectives to businesses, governments and stakeholders, on what is likely to be important or not, which allows them build future strategies for decision-making. To build future strategies for better decision-making, the exponential volume of data should be analyzed using automated and systematic methods. For example, in a business environment, using these methods provides institutions a better view of their customers' opinions, enabling them to develop their image or brand better.

Analyzing social networks data, or the so-called relationships between entities, can provide insight and awareness that is absent when considering the entities alone. Social network data are created from interactions between individuals, interactions amplified by personal relationships. These networks have interesting characteristics in terms of the data values. However, they have specific properties (power law distribution, small world, assortativity, preferential attachment, etc.) that require more sophisticated analysis tools that classical information systems approaches do not offer. For example, the community structure of social networks is one of the fundamental properties. The structure of these networks can be used to understand the interactions between entities but also to explain events. Indeed, observations have shown that some events emerge faster through social networks than through other traditional media including Websites, radio and television [31].

Recently, the analysis of social networks is focused on predictions and the effects of current strategies in the future, that are becoming a popular and a common interest to marketing, sales, and competitive intelligence analysts [10, 20]. Professionals dedicated to these strategies are able to analyse data that arise on social platforms to capture early signs of changes that might present opportunities like awareness and engagement, but even threats on the evolution of the environment [22]. Due to the arising volumes of these data, we are sometimes unable to see small significant clues that act as warnings of important events to come. Finding and capturing these clues is even harder if the events we are interested in are not known beforehand. Consequently, an adapted and automated system is required to exploit and process this data. Detecting early signs of change, often known as weak signals, allows policy and decision-makers to adapt anticipatory and more effective action strategies, rather than responding on the spot to the events as they happen. From here comes the need to find a method that relies on a graph of interactions between entities.

In this paper, we aim to present our approach in which we establish a method for identifying and interpreting weak signals. We hereby propose a method that relies on the network topology, by extracting particular network patterns, or so-called graphlets, that we consider an operational description of the weak signal. We propose to model the social data in the form of a temporal interaction graph between entities, and extract these patterns that could be indicators of important events in the future.

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08.
https://doi.org/10.1145/3548785.3548807

The rest of the paper is structured as follows: Section 2 presents some definitions of weak signals from the literature, and methods used to detect them in large data sets. In section 3, we introduce the proposed approach to identify and validate weak signals in social networks: we present a case study carried on a graph of temporal interactions between individuals. Section 4 describes an experimental setup to measure our algorithm's performance, then outlines the architecture of our approach consisting of different layers to support data processing and exploration. Finally, section 5 concludes the work presented in this paper and discusses future directions and perspectives.

## 2 BACKGROUND

Research on weak signals is largely influenced by the work of Ansoff in the 1970s [3]. He introduced a theoretical definition of the weak signal by considering it as a first symptom of strategic discontinuities acting as an early warning information, of weak intensity, which can announce a trend or an important event. In [4], Ansoff completes the signal's definition: a weak signal is a sudden, urgent, unfamiliar change in the firm's perspective which threaten either a major profit reversal or loss of a major opportunity. Depending on the authors and the domains, synonyms such as hunch, alarm signal have been proposed, but also adjectives associated with the signal, like early [6], critical [9], vague, etc. Indeed, weak signals can be revealed in many domains, from the detection of anomalies in a complex system like airplane management [1], to the protection of individuals such as the prevention of crimes [7] or harassment [18], but also in decision making and anticipation within the framework of the strategic planning of companies by using a Return on Experience (REX). By summarizing all of the proposed definitions in the literature review, we can define a weak signal as an **information that provides an indication of upcoming or emerging events that may have a significant impact on the system.**

Hiltunen's three-dimensional model [12] was proposed in the late 2000s. The author introduced the *importance* of a signal and the *graduation of a signal from weak to strong*, and also put forward the *rareness* of the signal. This model allowed, by highlighting the characteristics of weak signals, to provide an operational definition of this concept. We rely on this model in our approach. We translate the first two dimensions by the *support of the signal* which is a graph built from the temporal interactions, and a *phenomenon/cause*[1] which is an announcer of the event. As for the third dimension, *interpretation*, we rely on the expertise of decision-makers to make sense of the detected signals. Figure 1, inspired from Hiltunen's model, illustrates our perspective for these three dimensions.

In order to detect weak signals efficiently, Ansoff [4] suggested that they must pass three filters: 1) the monitoring (surveillance); 2) the mentality; and 3) the power, before potentially triggering an action or a decision. The monitoring filter relates to the capacity of identifying the weak signal in the midst of all other perceived information, by one or more actors within the organization. The mentality filter refers to the capacity of recognizing the signal once detected. Finally, the power filter refers to decision-making once the signal is detected and its relevance recognized. The people in charge in the organization can decide for example not to make this signal a priority, despite the underlying risk [29].

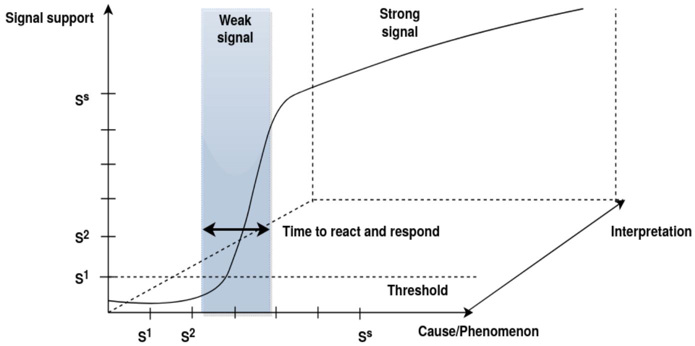


**Figure 1: Signal strengthening inspired by Hiltunen's three-dimensional model**

A survey [28] was done to present a theoretical background on the most employed methods and applications in the domain of weak signal detection. It should be noted that none of these methods offer to business experts specific tools for interpreting weak signals. These methods can be classified under several categories like statistics [14], graph theory [19] and machine learning [23]. As we have seen, the vast majority of these methods rely on keywords and documents analysis to identify some of them as weak signals, using text mining [17, 21, 30] and speech recognition. When dealing with data issued from social networks, applying the existing weak signals detection methods that are based on text mining or topic modeling for example, is not an easy task. The data issued from social platforms like Facebook or Twitter for example, consists of short texts (for Twitter they are up to at most 280 characters), containing abbreviations, spelling errors, special characters, urls, images, etc. In the following, we describe our proposed approach to detect and interpret weak signals in a temporal graph of interactions.

## 3 DETECTING WEAK SIGNALS USING NETWORK TOPOLOGY: A CASE STUDY WITH SOCIAL INTERACTIONS BETWEEN BABOONS

We rely on a topological analysis of social relations between entities, to extract patterns characterizing weak signals. We choose special network motifs, graphlets (first introduced in 2004 [26]), as an operational description to detect weak signals in a large graph of temporal interactions. Graphlets are induced subgraphs[2] connected and non-isomorphic[3], ranging from 2 to 5 nodes chosen among the nodes of a large graph. There are 30 different types going from $G_0$ to $G_{29}$[4]. An essential element in the context of graphlets are the orbits [25]. They represent the positions (or roles) occupied by the

[1] A phenomenon is an observed fact, normal or surprising event.

[2] In graph theory, an induced subgraph is a subset of the nodes and all their edges in the original graph.

[3] In graph theory, an isomorphism of two graphs G and H is a correspondence between the sets of nodes in G and H, such that if two nodes are adjacent in G, they are adjacent in H. Graphlets are non-isomorphic because they do not have the same shape.

[4] In this document, we use the term graphlet for each type among the 30, however this does not represent its occurrence.

nodes of these subgraphs. There are 73 different positions (from $O_0$ to $O_{72}$) for the 30 graphlets. Graphlets and their corresponding orbits are illustrated in figure 2.

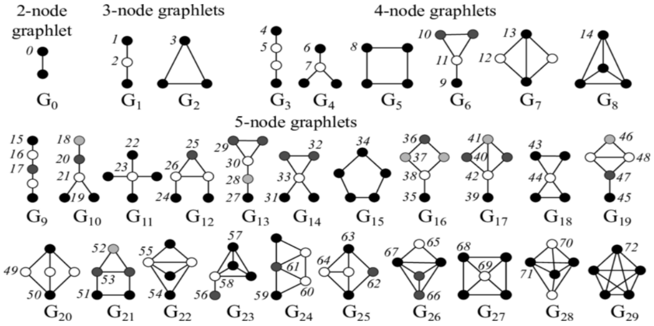


**Figure 2: 30 different graphlet types with their orbits, as introduced in [26].**

Indeed, we choose graphlets because they present characteristics generally associated with weak signals:

- They are **small patterns** consisting of few links between nodes;
- Some of them are **rare** in a large volume of information;
- They are however **interpretable** by business experts by means of their predefined shapes and orbits.

We aim to find a quantifiable property of weak signals based on graphlets, while describing a case study performed on a network representing a ground truth in a social network. The objective of this case study is to use the collected data, to identify and report indicators of changes in the behavior of individuals in contact, particularly those leading to an aggression. This identification can be seen as a source of opportunity to monitor the evolution of a social group over time, and probably prevent aggression between individuals of the group. We provide additional analysis components that help stakeholders and experts in interpreting and giving meaning to the identified indicators, which wipes out the "black box" effect that a fully automated approach could have. To this end, we start by describing the data, apply our method of detecting weak signals, and finally validate the resulted weak signals.

## 3.1 Data presentation

The raw data set used in this study represents a list of interactions between individuals belonging to a group of captive Guinea baboons [8]. We downloaded the corresponding raw files from the SocioPatterns Website that offers free online data sets to the scientific community[5]. The data span a time window of nearly a month between June and July 2019 and were collected by two different methods: 1) behavioral observations by trained human observers; and 2) a wearable sensor-based infrastructure.

To describe the latter infrastructure, the group of 20 baboons was fitted with leather collars. Two individuals are considered to be in contact during a 20-second interval, if their sensors have exchanged at least one packet during this interval, and the contact event is terminated when the sensors do not exchange any packets during a 20-second interval. We consider this infrastructure as the basis of the data used as an input to our detection of weak signals. It consists of 63,095 interactions in the form of a three-component tuple $(t, i, j)$, as shown in the extract of table 1: $t$ represents the timestamp[6] at which the interaction took place, $i$ and $j$ are the names of the individuals in contact.

| $t$ | $i$ | $j$ |
|---|---|---|
| 1560396500 | ARIELLE | FANA |
| 1560396500 | ARIELLE | VIOLETTE |
| 1560396520 | FANA | HARLEM |
| 1560396540 | FELIPE | ANGELE |
| 1560396540 | ARIELLE | FANA |
| 1560396580 | BOBO | FELIPE |

**Table 1: Extract from the raw file of baboon interactions, collected by the sensors.**

We are interested to find weak signals indicating a change in the behavior of individuals in contact, especially those leading to an aggression. Searching for weak signals in this context, offers perceptions of possible future situations that might be threats to the growth and the development of the society.

## 3.2 Identification of weak signals

Our aim is to describe weak signals with a signature in the form of a quantifiable property that characterizes them and helps with their detection amidst a large volume of data. We therefore use graphlets as an operational tool to establish this signature, characterized by the signal's visibility, diffusion, amplification and rareness. Table 2 provides a description of weak signals characteristics from conceptual and operational perspectives.

Since we are dealing with temporal interactions, we order the relations by $t$ and divide the original corpus into $s$ snapshots in order to study the diffusion and amplification of signals. A snapshot $S^i$ contains the nodes (baboons) and their relationships that occurred during the time interval $[i, i + \Delta t[$, with $\Delta t$ equal to 30 minutes, is the same duration of each snapshot. We applied our algorithms described below, for each snapshot. Algorithm 1 presents the steps followed in our approach, using graphlets for the identification of weak signals. First of all, 30-elements arrays are initialized to calculate velocities and accelerations in snapshot $S^i$, and store the results of event precursors in *candidates* and weak signals in *WS* (lines 2 and 3). For each snapshot $S^i$, each graphlet type $G_x$, $\forall x \in \{0, \ldots, 29\}$ is enumerated using the Orca algorithm[7] [13]. We choose Orca because it provides the exact enumeration of graphlets and orbits and it has an acceptable complexity. The result of Orca is stored in an array of 30 elements $G^i$ where $G^i[x]$ contains the number of the graphlet $G_x$ in the snapshot $S^i$ (line 4). The obtained values are then normalized using a procedure inspired from the work in [11], in which they study the similarity between two queries in

[5] http://www.sociopatterns.org/datasets/baboons-interactions/

[6] timestamp epoch: number of seconds elapsed since January 1, 1970 at 00:00; e.g. "13 June 2019 03:28:20" corresponds to timestamp 1560396500.

[7] https://rdrr.io/github/alan-turing-institute/network-comparison/src/R/orca_interface.R

| Dimension/Criterion | Conceptual definition | Operational definition |
|---|---|---|
| Visibility | Number/frequency | Number of graphlets in each snapshot |
| Diffusion | Velocity | Velocity of graphlets calculated w.r.t their number |
| Amplification | Acceleration | Acceleration of graphlets calculated w.r.t their velocity |
| Rareness | Contribution | Contribution of each graphlet w.r.t the number of all graphlets (ratio calculation) |

**Table 2: Conceptual and operational descriptions of weak signal criteria.**

**Algorithm 1:** Weak signals identification in a snapshot $S^i$

```
Inputs :$S^i$th snapshot, $\overline{G^{i-1}}$ normalized numbers of
        graphlets in snapshot $S^{i-1}$, $\overline{V^{i-1}}$ normalized
        velocities in snapshot $S^{i-1}$, real $k$
Output:Detected weak signals $WS$
1  begin
2      $WS \leftarrow \{\}$  $candidates \leftarrow \{\}$
3      $\overline{V^i} \leftarrow$ [NULL]; $\overline{A^i} \leftarrow$ [NULL] ;
4      $G^i \leftarrow$ Orca $(S^i, 5)$ ;  /* Count 5-nodes graphlets.
       */
5      for $x \leftarrow 0$ to $29$ i.e. for each type of graphlet $G_x$ do
6          $\overline{G^i[x]} \leftarrow$ Normalization $(x, i, G^i)$;
7          $\overline{V^i[x]} \leftarrow \overline{G^i[x]} - \overline{G^{i-1}[x]}$ ;          /* Velocity
            calculation for graphlet type $G_x$ */
8          $\overline{A^i[x]} \leftarrow \overline{V^i[x]} - \overline{V^{i-1}[x]}$ ;      /* Acceleration
            calculation for graphlet type $G_x$ */
9          if $\overline{V^i[x]} \geq k$ Or $\overline{A^i[x]} \geq k$ then
10             $candidates \leftarrow candidates \cup G_x$;
11         end if
12     end for
13     $WS \leftarrow$ Qualification $(G^i, candidates, k)$ ;
14     return $WS$
15 end
```

a temporal database (see algorithm 2 for details of the function). Even though the snapshots have same duration, we proceed by the normalization to re-scale graphlets number as the number of nodes and their corresponding links differ between snapshots (some snapshots consist of few links, and others are up to thousands of links).

From the normalized values, we calculate graphlets velocities and accelerations, that we use as quantifiable measures for the diffusion and amplification of signals. Based on these criteria, graphlets having velocity or acceleration values that are higher than a predefined entry threshold $k$, are selected among the candidates for weak signals (lines 6 to 12). To qualify the selected graphlets into weak signals, we consider the rareness criterion. The aim here is to quantify the contribution of each graphlet to the overall evolution of all graphlets using a ratio, to confirm whether they are weak signals or not (line 13). This function returns the list of graphlets, identified as weak signals.

**Algorithm 2:** Normalization Function

```
Inputs :Type of graphlet $G_x$, Graphlet numbers $G^i$ at all
        snapshots, number of all snapshots $s$
Output:Normalized value
1  begin
2      Calculate the mean of graphlet $G_x$ for $s$ snapshots:
        $\mu(G_x)$
3      Calculate the standard deviation of graphlet $G_x$: $\sigma(G_x)$
4      $Res \leftarrow \frac{G^i[x] - \mu(G_x)}{\sigma(G_x)}$ ;        /* Normalization */
5      return $Res$
6  end
```

**Algorithm 3:** Qualification function

```
Inputs :Graphlet number $G^i$ at snapshot $S^i$, candidates,
        real $k$
Output:List of weak signals
1  begin
2      $Res \leftarrow \{\}$ ;                    /* Initialization */
3      for $x \leftarrow 0$ to $29$ i.e. for each type of graphlet $G_x$ do
4          $R[x] \leftarrow \frac{G^i[x]}{\sum_{x=0}^{29} G^i[x]}$ ;   /* Contribution ratio
            calculation */
5      end for
6      while $Rank(R[x]) \leq k$ do
7          $arr_R \leftarrow arr_R \cup G_x$ ;            /* Choose top $k$
            contributions */
8      end while
9      if $G_x \in arr_R$ And $G_x \in candidates$ then
10         $Res \leftarrow Res \cup G_x$            /* True positives */
11     else if $G_x \in arr_R$ And $G_x \notin candidates$ then
12         $Res \leftarrow Res \cup G_x$            /* True negatives */
13     end if
14     return $Res$
15 end
```

Algorithm 3 describes the details of the qualification function. For each graphlet type we calculate the contribution ratios at a snapshot $S^i$ (lines 3 to 5), and rank the resulting values in ascending order. The contribution values that are less or equal than $k$ are chosen and stored in a list (line 7). Next, we apply the following checking rules that aim to maintain the true positives and negatives to be qualified as weak signals. If the graphlet is a candidate and its ratio is among the lowest contributions, then it is classified with the true positives, thus stored in the weak signals list. If the graphlet is a candidate but its ratio is not among the lowest contributions, then it is classified with the false positives or false alarms. If the graphlet is not a candidate but its ratio is among the lowest contributions, then it is classified with the true negatives, thus stored in the weak signals list. Finally, if the graphlet is not a candidate neither its ratio is among the lowest contributions, then it is classified with the false negatives. At the end of this step, we aim to maintain the true positives, add the true negatives, and eliminate the generated false alarms. So the algorithm returns only the list of true positives and true negatives, stored in the weak signals list (lines 9 to 13). Figure 3 displays a summary of the needed criteria and the used algorithms to identify weak signals in a snapshot $S^i$.

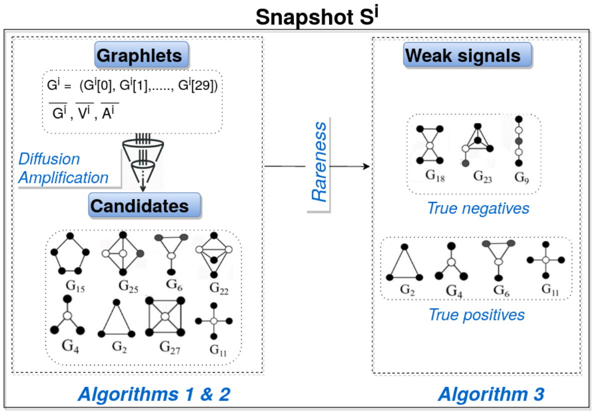


**Figure 3: Summary of the steps and algorithms used for the identification of weak signals at snapshot $S^i$.**

After applying these algorithms to the baboons data set, with respect to the paper's requirements, we present in table 3 the study carried on the single 08:00 a.m. snapshot of 19-06-2019.

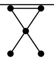
| **Graphlet** | $G_{25}$ | $G_{14}$ | $G_{27}$ | $G_7$ | $G_{11}$ |
|---|---|---|---|---|---|
| **Shape** | |  | | | |
| **Contribution** | 0.0198 | 0.0246 | 0.0324 | 0.0360 | 0.0381 |

**Table 3: Top 5 graphlets qualified as weak signals in the 8:00 a.m. snapshot.**

## 3.3 Validation of identified signals

We use the observations recorded by a human to validate the weak signals that we have identified in the preceding subsection. Behavioral observations were conducted for 5 days per week (between June 13 and July 10, 2019) using the focal sampling method [2], with two sessions of approximately two hours per day at different times each day, ranging from 8:00 a.m. to 5:00 p.m. During each session, a trained observer focused on each individual for a period of 5 minutes and recorded their behaviors. The data file recorded by this observer contains 5377 interactions, and it is composed of seven columns detailed below:

**DateTime:** The timestamp of the interaction, i.e. the moment of the recording of an action;

**Actor:** The name of the baboon;

**Recipient:** The name of the baboon on whom the actor acts;

**Behavior:** The behavior of the actor. There are 15 different types of behavior including 'Rest', 'Play with', 'Grunt-chew', 'Beg', 'Threaten', 'Submit', 'Touch', 'Avoid', 'Attack';

**Category:** The classification of behaviors. A behavior can be 'Affiliated', 'Agonistic' or 'Other';

**Duration:** in seconds, of the observed behavior, One-off contacts have no duration;

**Point:** Indicates if the contact is a POINT event (YES) or a STATUS event (NO).

We placed ourselves on the same period that we have studied in the sensor data set. Table 4 shows an extract of the observed data by the human on 19-06-2019 between 08:58 and 08:59 a.m., where we noticed a transition from affiliative to agonistic behaviors. As of the third row of this table, the behavior becomes agonistic between individuals who were interacting quietly a second before (LOME and FELIPE), and at 09:11, their interaction returns affiliative. In the same data set, we noticed that these agonistic behaviors were followed by attacks between VIOLETTE, MUSE, HARLEM and MALI at 09:17 a.m.

| DateTime | Actor | Recipient | Behavior | Category | Duration | POINT |
|---|---|---|---|---|---|---|
| 19/06/2019 08:58 | LOME | FELIPE | Resting | Affiliative | 17 | NO |
| 19/06/2019 08:58 | LOME | ANGELE | Resting | Affiliative | 17 | NO |
| 19/06/2019 08:59 | FELIPE | LOME | Submission | Agonistic | 0 | YES |
| 19/06/2019 08:59 | ANGELE | LOME | Submission | Agonistic | 0 | YES |
| 19/06/2019 08:59 | ANGELE | LOME | Attacking | Agonistic | 0 | YES |
| ........ | ..... | ...... | ...... | ..... | ... | ... |
| 19/06/2019 09:17 | VIOLETTE | HARLEM | Chasing | Agonistic | 0 | YES |
| 19/06/2019 09:17 | VIOLETTE | HARLEM | Submission | Agonistic | 0 | YES |
| 19/06/2019 09:18 | ANGELE | VIOLETTE | Threatening | Agonistic | 0 | YES |
| ........ | ..... | ...... | ...... | ..... | ... | ... |

**Table 4: Extract from the data recorded by the human observer, in the morning of 19-06-2019.**

We took back the sensor data set and we moved to a finer analysis of graphlets, to identify the baboons appearing in weak signals graphlets. The identified baboons are those who participated in the aggression reported by the human observer one hour later.

To do this, we implemented *Cypher* queries in a *Neo4j* graph database, to select a particular graphlet instance and its belonging nodes. Considering the collected sensor interactions which took place on 19-06-2019 at 08:00 a.m., figure 4 shows an extract of particular instances of the graphlets weak signals (listed in table 3), in which we find the individuals mentioned above, in positions that are sometimes central (e.g. FELIPE in a), and other times peripheral (e.g. FELIPE in b).

Listing 1 is an example of the *Cypher* query that returns a $G_{27}$ instance, after specifying the labels of nodes belonging to this instance (because multiple nodes can occupy the same orbit in different instances). In this instance, ARIELLE occupies the central orbit $O_{69}$, and the remaining individuals notably FELIPE, HARLEM, FANA and VIOLETTE occupy the peripheral orbit $O_{68}$.

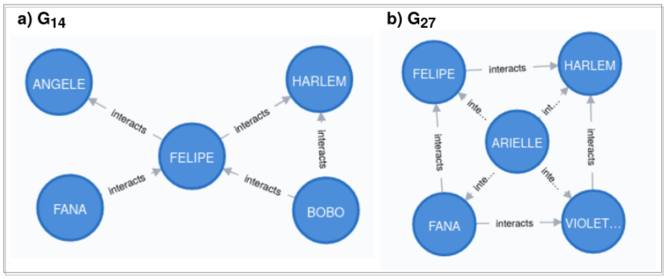


**Figure 4: Baboons in particular instances of weak signal graphlets at 08:00 a.m.**

```
MATCH (u1)--(u2)--(u5),(u2)--(u3)--(u5),(u3)--(u4)--(u5),(u4
    )--(u1)--(u5)
WHERE NOT ((u1)--(u3)) AND NOT ((u2)--(u4))
  AND u1.name = 'FANA' AND u2.name = 'VIOLETTE'
  AND u3.name = 'HARLEM' AND u4.name = 'FELIPE'
  AND u5.name = 'ARIELLE'
RETURN *
```

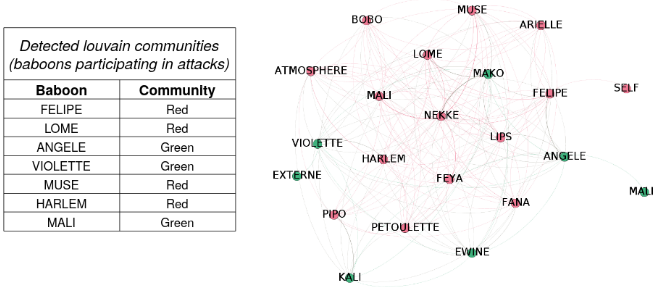
**Listing 1: Example of a Cypher query that returns an instance of $G_{27}$**

In this step, we allowed experts to give sense to the detected weak signals, by providing contextual elements that reveal the different roles occupied by nodes belonging to the orbits of these signals. These elements enable them focus their interpretation on particular individuals that might be the cause of a future aggression behavior, as we have discovered after comparing our results with the human observations.

### 3.4 Study of community structure

Identifying community structure in social networks is important since it helps in understanding the topological interactions of individuals in the network, as well as discovering their shared information. At night, the whole group of baboons gather to shelter themselves from predators in the trees, during the day they divide into small groups. So, we observed the distribution of interactions between individuals, on the level of the global graph, all snapshots together. We ran the Louvain algorithm [5] on the global graph of the sensor corpus to reveal communities of nodes[8] as illustrated in figure 5, and differentiated by red and green colors as shows the side table of the figure. The community in red is the one that contains most of the individuals from the group. Indeed this is normal because the individuals who are in it, notably FELIPE, LOME and HARLEM, have participated the most in the interactions (whether they are affiliative or agonistic). We also found that there are important links between individuals belonging to different communities, like FELIPE and ANGELE, or VIOLETTE and HARLEM for example.

The community detection algorithm is not sufficient to prevent attacks, since the attacks exist between individuals of the same community, as they exist between two different communities.



*Detected louvain communities (baboons participating in attacks)*

| Baboon | Community |
|---|---|
| FELIPE | Red |
| LOME | Red |
| ANGELE | Green |
| VIOLETTE | Green |
| MUSE | Red |
| HARLEM | Red |
| MALI | Green |

**Figure 5: Two communities detected by the Louvain algorithm.**

We considered in BEAM an exploration of weak signals via quantitative and qualitative features. We applied a refined study on specific nodes' properties (example their type, their position in the graphlets, their community) to uncover their structural relationships if they tend to regroup together or not, and how much they contribute in the identified weak signals by the method.

## 4 IMPLEMENTATION DETAILS

In this section, we discuss the detailed implementation of our method, as well as the architecture that illustrates our proof of concept for weak signal identification and interpretation method. We first present a study on the behavior of the graphlets counting algorithm Orca, with respect to reasonable graph sizes processed by snapshots, on which we relied to measure the performance and the response time of the algorithm.

### 4.1 Orca running time analysis

There exist several algorithms to enumerate graphlets and orbits of a graph [27]. To choose the most convenient algorithm for counting graphlets and orbits in the studied graph structures, we have defined 3 essential criteria: 1) exact counting of graphlets that are up to five nodes, to maintain the interpretability of the results; 2) orbits counting for the study of nodes positions within each graphlet; and 3) availability of source code. We relied on the Orca algorithm proposed by Hočevar and Demšar in 2014 [13], which is an exact counting algorithm, coming from an analytic approach based on matrix representation, and works by setting up a system of linear equations per node of the input graph that relate different orbit frequencies. Theoretically, Orca can operate on k-node graphlets with $2 \leq k \leq 5$. On one hand, the computation cost grows dramatically as k increases. On the other hand, it is easier to explore the graph when k is small. Considering $e$ as the number of edges and $d$ the maximum degree of nodes, its time complexity is of $O(ed)$ for four-node graphlets and $O(ed^2)$ for five-node graphlets.

We performed an experimental analysis to evaluate Orca's implementation complexity. The experiment consists of two possibilities, by: 1) fixing the number of nodes and increasing the number of links; or by 2) fixing the number of links and increasing the number of nodes accordingly.

In the first possibility, we generated random graphs after fixing a small number of nodes equal to 200, and increasing the number of links starting by 50% of the nodes number. Figure 6 represents the

[8] Community structure is one of a network's characteristics where nodes belong to dense connected sub groups of this network.

response time of Orca according to the measured density of each of the considered graphs. In this figure we notice that there are thresholds. Whenever the density of the graph is lower than 0.4, the time consumption is less than 10 seconds. For a density between 0.4 and 0.6, the response time increases from 10 to 40 seconds (as if the time is multiplied by four). After a density of 0.85 the response time is almost multiplied by two, then remains stable.

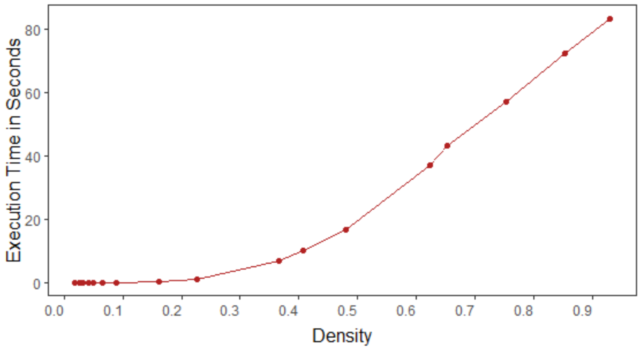


**Figure 6: Orca’s response time according to 200 nodes graphs and an increasing density.**

As for the second possibility, we designed an experimental set to measure the performance of the algorithm, while fixing the number of links in the graphs and varying the number of nodes to reach 80% of the number of links. After applying BEAM on several data sets of different sizes, roughly the standard graphs for estimating the behavior of Orca must have an average of 5.000 links. Therefore we designed the experiment according to a fixed number of links 2.500, 5.000, 10.000 and 100.000 respectively. Table 5 contains an extract of the properties for the graphs used in the experiment, with links fixed to 5.000 and 10.000 respectively, along with the corresponding elapsed time of Orca in milliseconds. We highlighted the highest values for Orca’s response time in red, and the lowest values in blue. This extract confirms well the theoretical behavior of the algorithm in terms of an increased response time w.r.t an increased graph density. It is to be noted also that the algorithm remains within reasonable times to process data of a snapshot on real graphs.

## 4.2 Architecture description

To meet the needs in terms of performance, interoperability (use of third party tools to apply complementary analysis to help end-users to interpret graphlets) and adequacy between the data structures handled by the different algorithms described above, we have specified an architecture representing the approach with three layers. These layers correspond to: the 1) data storage; 2) the detection; and 3) the interpretation of weak signals, which communicate through third-party tools. Figure 7 illustrates the specified architecture for our approach.

The storage layer contains the raw data collected from the various sources, as well as the resulted data from the two other layers. This layer supports different types of source data files as in CSV or TXT formats, or in JSON format (for example the tweets are downloaded from the Twitter API in the form of JSON files[9]). The resulted data from the other two layers, can be stored in two types of database management systems. The first system is relational, for which we use the PostgreSQL database, and the second one is graph-based, for which we use the Neo4j database.

| Links | Nodes | Density | Elapsed Time (in milliseconds) |
|---|---|---|---|
| 5.000 | 500 | 0.04 | 118 |
| | 600 | 0.027 | 85 |
| | 1.500 | 0.004 | 23 |
| | 2.400 | 0.002 | 16 |
| | 3.500 | 0.0008 | 14 |
| | 4.000 | 0.0006 | 13 |
| 10.000 | 1.000 | 0.02 | 258 |
| | 1.500 | 0.009 | 110 |
| | 3.300 | 0.002 | 45 |
| | 5.200 | 0.0007 | 32 |
| | 7.000 | 0.0004 | 29 |
| | 8.000 | 0.0003 | 28 |

**Table 5: An extract of the experiment carried on to measure Orca’s behavior according to a list of fixed number of links.**

The detection layer consists of three steps: 1) raw data processing; 2) candidates selection; and 3) weak signals qualification. This layer makes use of third-party tools including R `igraph` graphs library, and Orca algorithm for graphlets counting. It sends the detected weak signals to the upper layer, to be interpreted.

The interpretation layer contains all possible components that give sense of the detected signals in the lower layer, hence help an expert to determine their relevance for future planning. These components include first the identification of nodes positions within weak signals graphlets, thanks to their orbits. These positions are provided by snapshot, by node and by orbit (they represent central, intermediary and peripheral positions). The components consist also of centrality and community measures applied at the nodes level. In addition, this layer includes visualization methods that give experts insights about the distribution of most important nodes for example in weak signal graphlets. They consist of (but are not limited to) Sankey diagrams, histograms, pie charts and heatmaps. These visualizations can be given at the level of an individual snapshot, and for all snapshots combined. Finally, this layer offers semantic analysis through an examination of the nodes characteristics, their activity and their role in the society.

These three layers communicate via reproducible workflows that we implemented using Jupyter notebooks [24]. Jupyter notebook is a tool used for data exploratory analysis. It allows data scientists to create scripts combining code, text and graphical interfaces. Specific kernels for different programming languages run independently and interact with Jupyter, including Python, R, and Scala. We consider the Jupyter notebook as an interactive environment that allows us to gather a description of the input data, develop with multiple programming languages in a single kernel, and then save and convert the results to formats other than structured JSON files,

[9] https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview

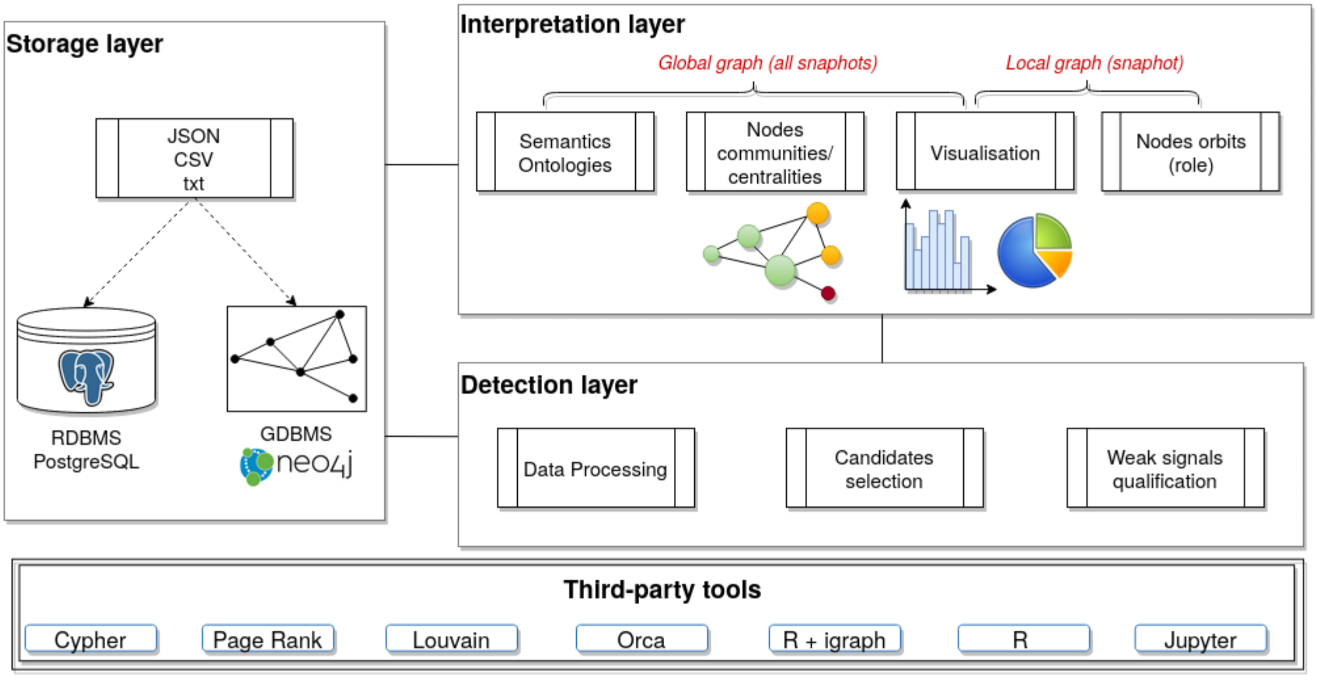
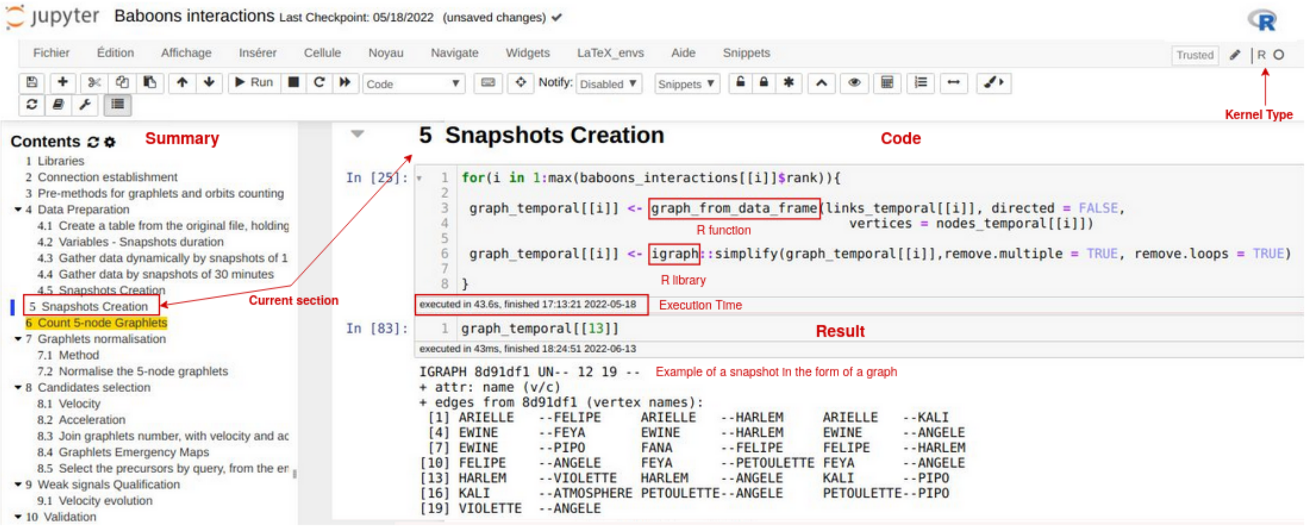


**Figure 7: Architecture of the platform consisting of three layers.**



1 Libraries
2 Connection establishment
3 Pre-methods for graphlets and orbits counting
4 Data Preparation
4.1 Create a table from the original file, holding
4.2 Variables - Snapshots duration
4.3 Gather data dynamically by snapshots of 1
4.4 Gather data by snapshots of 30 minutes
4.5 Snapshots Creation
5 Snapshots Creation
6 Count 5-node Graphlets
7 Graphlets normalisation
7.1 Method
7.2 Normalise the 5-node graphlets
8 Candidates selection
8.1 Velocity
8.2 Acceleration
8.3 Join graphlets number, with velocity and ac
8.4 Graphlets Emergency Maps
8.5 Select the precursors by query, from the er
9 Weak signals Qualification
9.1 Velocity evolution
10 Validation

5 Snapshots Creation

In [25]:
```
for(i in 1:max(baboons_interactions[[i]]$rank)){

 graph_temporal[[i]] <- graph_from_data_frame(links_temporal[[i]], directed = FALSE,
                                               vertices = nodes_temporal[[i]])

 graph_temporal[[i]] <- igraph::simplify(graph_temporal[[i]],remove.multiple = TRUE, remove.loops = TRUE)
}
```

executed in 43.6s, finished 17:13:21 2022-05-18

In [83]:
```
graph_temporal[[13]]
```

executed in 43ms, finished 18:24:51 2022-06-13

```
IGRAPH 8d91df1 UN-- 12 19 --
+ attr: name (v/c)
+ edges from 8d91df1 (vertex names):
 [1] ARIELLE   --FELIPE     ARIELLE   --HARLEM     ARIELLE   --KALI
 [4] EWINE     --FEYA       EWINE     --HARLEM     EWINE     --ANGELE
 [7] EWINE     --PIPO       FANA      --FELIPE     FELIPE    --HARLEM
[10] FELIPE    --ANGELE     FEYA      --PETOULETTE FEYA      --ANGELE
[13] HARLEM    --VIOLETTE   HARLEM    --ANGELE     KALI      --PIPO
[16] KALI      --ATMOSPHERE PETOULETTE--ANGELE     PETOULETTE--PIPO
[19] VIOLETTE  --ANGELE
```

**Figure 8: Jupyter notebook of the case study realized on baboons interactions.**

such as HTML and PDF. Figure 8 shows a capture of the Jupyter notebook editor representing the described case study in this paper, baboons interactions. On the top right of this figure, the type of the kernel used to compile the written code is displayed. As mentioned earlier, we use the R kernel to execute our scripts. On the left there is a navigation panel that shows the sections that compose this notebook. The content of the highlighted section (5) in yellow in this panel, is shown in the center cell of the editor. The scripts are implemented using R libraries and functions, to create the series of snapshots (in the form of temporal graphs) from the baboons interaction data. Once the execution of the R scripts in this cell is finished, the execution end-time is automatically displayed under the cell. Below the center cell is another one representing the resulted graph of a particular snapshot. Here the number of the snapshot is 13, so the result shows the list of nodes interacting in this snapshot.

Apart from the reproducibility that Jupyter notebook offered to our approach, their interactive platform eased the exploratory data analysis. It also provided us a way to craft a story with the processed data in the different layers of this architecture.

## 5 DISCUSSION AND CONCLUSION

We presented in this paper our proposed approach to identify weak signals in social networks, by choosing graphlets as an operational description. We first find graphlets in a temporal interactions graph, quantifiable using signal diffusion and amplification that characterize them as candidates. Then, we measure the contribution of these graphlets, to qualify the true positives and true negatives, and identify the false alarms. An additional step is performed in which we use the predefined shapes and orbits of the graphlets, to help experts in interpreting the discovered signals.

We applied our approach on a ground truth data set representing social interactions between a group of Guinea baboons, and we were able to detect weak signals. By comparing our results with those recorded by a human observer of the baboons interactions, we confirmed that the detected signals appear prior to an aggression behavior, that was reported one hour later by the human observer. These results must be shared with the stakeholders and the persons in charge, to provide better control on such behavior, and perhaps build a preventive strategy in the future. Data and experimental programs are available under https://github.com/hibaaboujamra/Weak-Signals-Detection-and-Interpretation-BEAM. Our method has also been validated on other Twitter datasets [15, 16].

However, our approach still presents few limitations related to the filters introduced by Ansoff (detailed in section 2) that hinder the analysis of weak signals and constitute barriers to their interpretation. These limitations are mainly linked to:

(1) the constitution of the study corpus (monitoring filter);
(2) the interpretation of the detected signals through their recognition (mentality and power filters).

The first limitation can be settled by adding a feedback loop to allow business experts modifying the data selection filters, if for them, the discovered signals are not relevant. The second one depends highly on the expert's sufficient knowledge about the utility of the discovered signals.

We are currently addressing the mentioned limitations of our method mainly to ensure controlling the data selection filters, and the covering of different detection scenarios. In further research, we would like to enlarge our scope and explore the use of clustering techniques for the discovery of groups and profiles of nodes in the studied temporal graphs.

## ACKNOWLEDGMENTS

This work is supported by the "Investissements d'Avenir" program, Cocktail ISITE-BFC project (Initiatives Science Innovation Territoires Économie en Bourgogne-Franche-Comté), ANR contract 15-IDEX-0003, https://projet-cocktail.fr/.

## REFERENCES

[1] J. L. Ackley, T. G. Puranik, and D. Mavris. A supervised learning approach for safety event precursor identification in commercial aviation. In *AIAA Aviation 2020 Forum*, page 2880, 2020.

[2] J. Altmann. Observational study of behavior: sampling methods. *Behaviour*, 49(3-4):227–266, 1974.

[3] H. I. Ansoff. *Managing surprise and discontinuity: strategic response to weak signals.* European Institute for Advanced Studies in Management, 1975.

[4] H. I. Ansoff and E. J. McDonnell. Implanting strategic management, 1990.

[5] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[6] B. Coffman. Weak signal research, part I: Introduction. *Journal of Transition Management*, 2(1), 1997.

[7] T. Davies and E. Marchione. Event networks and the identification of crime pattern motifs. *PloS one*, 10(11):e0143638, 2015.

[8] V. Gelardi, J. Godard, D. Paleressompoulle, N. Claidière, and A. Barrat. Measuring social networks in primates: wearable sensors versus direct observations. *Proceedings of the Royal Society A*, 476(2236):20190737, 2020.

[9] M. Godet. *From anticipation to action: a handbook of strategic prospective.* UNESCO publishing, 1994.

[10] K.-Y. Goh, C.-S. Heng, and Z. Lin. Social media brand community and consumer behavior: Quantifying the relative impact of user-and marketer-generated content. *Information systems research*, 24(1):88–107, 2013.

[11] D. Q. Goldin and P. C. Kanellakis. On similarity queries for time-series data: constraint specification and implementation. In *International Conference on Principles and Practice of Constraint Programming*, pages 137–153. Springer, 1995.

[12] E. Hiltunen. The future sign and its three dimensions. *Futures*, 40(3):247–260, 2008.

[13] T. Hočevar and J. Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, 12 2014.

[14] J. Huang, M. Peng, H. Wang, J. Cao, W. Gao, and X. Zhang. A probabilistic method for emerging topic tracking in microblog stream. *World Wide Web*, 20, 03 2017.

[15] H. A. Jamra, M. Savonnet, and É. Leclercq. Detection of Event Precursors in Social Networks: A Graphlet-Based Method. In S. S. Cherfi, A. Perini, and S. Nurcan, editors, *Research Challenges in Information Science - 15th International Conference, RCIS 2021, Limassol, Cyprus, May 11-14, 2021, Proceedings*, volume 415 of *Lecture Notes in Business Information Processing*, pages 205–220. Springer, 2021.

[16] H. A. Jamra, M. Savonnet, and É. Leclercq. BEAM: A Network Topology Framework to Detect Weak Signals. *International Journal of Advanced Computer Science and Applications*, 13(4), 2022.

[17] H. Kim, S.-J. Ahn, and W.-S. Jung. Horizon scanning in policy research database with a probabilistic topic model. *Technological Forecasting and Social Change*, 146:588–594, 2019.

[18] H. Kim, Y. Han, J. Song, and T. M. Song. Application of social big data to identify trends of school bullying forms in south korea. *International journal of environmental research and public health*, 16(14):2596, 2019.

[19] L.-N. Kwon, J.-H. Park, Y.-H. Moon, B. Lee, Y. Shin, and Y.-K. Kim. Weak signal detecting of industry convergence using information of products and services of global listed companies-focusing on growth engine industry in south korea. *Journal of Open Innovation: Technology, Market, and Complexity*, 4(1):10, 2018.

[20] X. Luo, J. Zhang, and W. Duan. Social media and firm equity value. *Information Systems Research*, 24(1):146–163, 2013.

[21] J. Maitre, M. Ménard, G. Chiron, A. Bouju, and N. Sidère. A Meaningful Information Extraction System for Interactive Analysis of Documents. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 92–99, 2019.

[22] B. A. Miller, M. S. Beard, and N. T. Bliss. Eigenspace analysis for threat detection in social networks. In *14th International Conference on Information Fusion*, pages 1–7. IEEE, 2011.

[23] Y. Ning, S. Muthiah, H. Rangwala, and N. Ramakrishnan. Modeling Precursors for Event Forecasting via Nested Multi-Instance Learning. KDD '16, page 1095–1104, New York, NY, USA, 2016. Association for Computing Machinery.

[24] F. Perez and B. E. Granger. Project jupyter: Computational narratives as the engine of collaborative data science. *Retrieved September*, 11(207):108, 2015.

[25] N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

[26] N. Pržulj, D. G. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004.

[27] P. Ribeiro, P. Paredes, M. E. Silva, D. Aparicio, and F. Silva. A survey on subgraph counting: concepts, algorithms and applications to network motifs and graphlets. *arXiv preprint arXiv:1910.13011*, 2019.

[28] P. Rousseau, D. Camara, and D. Kotzinos. Weak signal detection and identification in large data sets: a review of methods and applications. May 2021.

[29] B. L. van Veen, J. Roland Ortt, and P. Badke-Schaub. Compensating for perceptual filters in weak signal assessments. *Futures*, 108:1–11, 2019.

[30] J. Yoon. Detecting weak signals for long-term business opportunities using text mining of web news. *Expert Systems with Applications*, 39(16):12543–12550, 2012.

[31] F. Zarrinkalam and E. Bagheri. Event identification in social networks. *Encyclopedia with Semantic Computing and Robotic Intelligence*, 1(01):1630002, 2017.

# Meta-stasis of the Internet

Bipin C. Desai
Concordia University
Montreal, Canada
BipinC.Desai@concordia.ca

## ABSTRACT

This paper offers a brief history of the information age in order to demonstrate how the loss of user control and the increase in certain forms of automation have metastasized into imminent and ongoing threats to social order and the democratic way of life.

The internet was established after a number of developments which included the interconnection of computers without extensive need of action by the users. It led to the introduction of user communication sub-systems such as text, email, file sharing and systems for searching for files. The so called information age is said to be marked by the adaption of a hypertext transport protocol in the last decade of the twentieth century. The information age was marked by a number of meetings which included the first of the world wide web conference in April 1994 followed by the second (Oct. 1994) and the third(April 1995) in quick succession. Other, by invitation only, meetings which dealt with issue of this era were held in Denver, OH(Metadata) and (America in the Age of Information)Bethesda, MD.

However, in just under three decades this information age has *meta*-stasis-ed into a form that is a threat to our social order and democratic way of life while fostering division. Enormous wealth has been garnered by just a few corporations and individuals at the expense of the harm it is doing to people all over the globe. This is the result of the spreading of fake-news and favouring angry content that result in civil strife and loss of lives. It has led to divisiveness and autocratic governments. Some so called democracies are in name only with the same people continuing in their 'elected' position from term to term, ad infinitum. Just as in the metastasis of a cancer, until it is checked, this transformed internet will destroy some vital parts of our everyday existence: our privacy and liberty while promoting an inegalitarian spirit.

## CCS CONCEPTS

• **General and reference**; • **Software and its engineering**; • **Social and professional topics**; • **Applied computing**; • **Security and privacy**;

## KEYWORDS

Online Social Networks, personal data exploitation, privacy, security, smartphones, tracking, big tech, lack of legislative control, surveillance

**ACM Reference Format:**
Bipin C. Desai. 2022. Meta-stasis of the Internet. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3548785.3548805

## 1 INTRODUCTION

Humans have evolved over the millenniums, however, it is only in the last few centuries that history has been recorded in a form that could be easily reproduced and transmitted from generation to generation. This was made possible with the invention of the printing press, widely considered to be a revolution. Humans have gone through a number of revolutions: roughly, the replacement of one type of social order by another.

Most revolutions are periods in a nation when one system is replaced with another: the replacement is to oust an autocratic government with something more representative[6]. Some are successful, others not so. The revolution of the North American British colony was replaced by a democratic federal system with a constitution whose interpretation and rigidity has and continues to create problems. The Chinese, French and Russian revolutions replaced one tyranny by another after wars, suffering, and tens of thousands of fatalities.

Here we are not discussing these revolutions but those which have been called industrial revolutions through multiple generations of digital computing devices, and development of the internet from connecting a number of computers with a small number of users to connecting millions of computers and billions of mobile devices as clients. We briefly talk about the first two and then turn to the subject of this paper namely the *meta*stasis of the internet.

## 2 THE INDUSTRIAL REVOLUTIONS

It took thousands of years before the mechanical advantage offered by simple machines were combined ingeniously into more complex machines. The power source to drive these machines were either human or animal. Other sources such as wind was also harnessed over the years. However, a more reliable one was needed. Boiling water to show the property of steam was developed over two thousand years ago and early versions of the steam engine were invented in the sixteenth century. It became the driving force of the first industrial revolution, said to be from 1760 through 1850.

The topic of the industrial revolutions is a vast one and is covered extensively in many sources[3]. The first industrial revolution, was in an era when industrial ownership and its products created a class of wealthy entrepreneurs, It also meant a shift from rural to

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548805

urban environments. Working in badly configured factories was a gruelling experience. Workers were not allowed free time and worked long hours; the work force included children and the pay was dismal.

During this period radical changes impacted not only agriculture but transportation and the social structure. The use of steam engines in mills and factories and subsequently in railways was an achievement of this period. The industrial revolution created a new source of great wealth for a handful of entrepreneurs, the owners of the factories while exploiting the workers including child labour. The industrial revolution led to an exodus of workers from agricultural, rural settings to city slums with health issues[19].

In the first industrial revolution the source of energy was coal. Some historians quibble over the exact boundary between the first and the second industrial revolutions, that started around the mid-19th century. A primary difference is that the second was the beginning of mass production in manufacturing and consumer goods. The power sources for the second industrial revolution were oil and gas, and the internal combustion engine. One can also mention the third industrial revolution wherein the power source was electrical and subsequently nuclear energy, electric motors, assembly lines in automated factories. It has been followed by a revolution, the fourth one, in which digital communication technology and the internet changed how information and interaction are managed.

## 3 THE DIGITAL GENERATIONS

The idea of a programmable computer is often traced to Charles Babbage[13]; he was a mathematician, philosopher, inventor and mechanical engineer. (In a recent book[77], one author seems to give this honour to the Majorcan polymath Ramon Llull who designed a machine made of paper.) Humans had to wait another century before the vision of Babbage was realized, in the form of digital computers, first using electro-mechanical components and subsequently all electronic components.

The centre part of today's computing and communication technology including the internet and mobile phones is the need for digital devices and infrastructure. The first generation of digital computer systems, which replaced analog systems, were devices made from vacuum tubes. They were large in size and required a considerable amount of electrical energy. They were relatively slow and had very limited storage capacity.

The next generation of digital computing devices made use of solid state devices using diodes and transistors. Each such device was distinct; the advantage was lower power requirements and size.

The third generation of digital systems used integrated circuits and hence there was considerable reduction in size and power needs. Time sharing and remote access as illustrated here was possible. Initially, the communication system was an analog acoustic coupler and the input output was an updated teletype. A higher level of integrated circuit allowed more powerful digital computers and development in operating systems along with features including multi-tasking and multi programming. These features allowed time sharing a central computer by many users using a dedicated high-speed telecom line.

The emergence of large scale integration and its refinement and miniaturization led to the mini and microcomputers, personal computers, laptops, notebooks, tablets and smartphones. This is the fourth generation of digital computers. These could be called the fourth generation of digital devices. There was speculation about the fifth generation of computers, however, with the advent of cloud computing and a few super computers, the focus of the fifth generation has more or less dropped out of sight with the emergence of cloud computing which, ironically, returns to the earlier generation as a form of time-sharing++!

## 4 THE INTERNET AND WORLD WIDE WEB

As more powerful computers were introduced in the early 1960s, organizations including universities set up a central computer centres to house powerful systems. A user who wanted to execute her program was required to prepare the program using punched cards and bring the deck of these cards to the centre. Common programming languages used in the early 1960s were FORTRAN and COBOL. The program, along with any data would be in a deck of cards which would be submitted to run in batches of programs written in the same lanuguage. Once the submitted program for a user is run, its output would be printed and both the output and the deck of cards would be picked up by the users. The procedure, if required, needed to be repeated for any changes to the program or data!

Later, with improvement of the operating systems and the introduction of telecommunication facilities and an acoustic coupler it was possible to use a remote terminal to input the program and run it. This would be feasible for small programs but larger programs needed the previously mentioned manual procedure: however, satellite stations with a mini-computer, could be set up to avoid trips to the computer centre. These satellite stations were connected to the central computer with a high-speed, dedicated telecommunication line—-this was the advent of connecting computers! An example of this was the use of the central computers located in a downtown campus by users in a suburban campus[1].

Systems such as time-sharing in early 1960s allowed many users to log into a central system from remote terminals, and store and share files on the central disk. Messaging between users of the same system also became feasible. Computer-based messaging between users of the same system became possible following the advent of time-sharing in the early 1960s

The introduction of the third generation computers in the mid 1960's and the use of time-sharing was the period which prompted, in May 1964, MIT professor Martin Greenberger to write the following in an article in The Atlantic[42]:

"Computing services and establishments will begin to spread throughout every sector of American life, reaching into homes, offices, classrooms, laboratories, factories, and businesses of all kinds."

The earliest form of email was introduced on a Unix system, in the early 1970s: this allowed users to compose a message and send it to the mailbox of other users on the system. With the interconnection of computer systems over the early network such as ARPANET in the early 1980s. There were a number of dedicated networks of interconnected computers in the world until the adoption of

[1] The students and faculty members of Loyola College used the central computer located at the downtown McGill University in the late 1960s - early 1970s.

**Figure 1: Remote computing**

***The author using an acoustic coupler to communicate with a remote main frame -early 1970s***

the packed-based communication (TCP/IP) of digital information, standardized in 1982. This led to the emergence of a world-wide network of fully interconnected networks all using the protocol[IP]. Initially, the internet connected systems at a number of organizations (including universities) and was accessed by people at those institutes to communicate and share[91]. The Simple Mail Transfer Protocol (SMTP) protocol was introduced[90] to send mail from a user on one computer to a user on a remote computer.

The web came into existence in the late 1980s with the development of a hypertext transmission protocol(HTTP)[94], an application of the TCP/IP protocol, and the first text browsers supporting the early Hypertext Markup Language(HTML)[93] standard. With the introduction of the graphical browser in the early 1990s, data sharing was for the first time extended to the masses[94].

As noted in [20] "even before the introduction of the web, the internet had made it possible for people to communicate via electronic mail (email)[69],[88] and on-line chat, allowed sharing of files[87] using anonymous file transfer protocol (FTP), news (Usenet News), remote access of computers (telnet), Gopher (a tool for accessing internet resources), Archie (a search engine for openly accessible internet files) and Veronica (search for gopher sites). These early systems afforded the opportunity of interconnecting people (who wanted to be connected), sharing resources without requiring anything in return and providing security and privacy; there was not yet any question of monetizing; the whole concept was to share without exploitation or expropriation of user data or content. However, these systems were not adopted widely: a key limitation of these early internet tools was the need to have some computing savvy; another challenge was the lack of an infrastructure to transfer the know-how to novices. This was also a limitation for the early web with the use of user unfriendly, text-based web browsers and a lack of training facility and easy to learn tools to build and maintain hypertext documents.

**Figure 2: WWWI Navigation Workshop**

***The author and colleagues during the WWW I - Navigation workshop in 1994 -in this forum, the author put forward the ideas of web history and search engine.***

Some early attempts to create software for hypertext[98] were buried by the emergence of the early tech giants who were more interested in having their system dominate the internet and limiting users from learning the basics. This strategy of dumbing down is behind all current systems, which has contributed to a downgrading of literacy and replacement of reading by videos and sound clips.

## 4.1 Web of Big-techs

The web was quickly recognized by business interests as an opportunity for commercial exploitation and this led to an explosion in the creation of data. The first few meetings of the World Wide Web conferences, for example the ones in Geneva (May 1994), Chicago (Oct. 1994) and Darmstadt (Apr. 1995), were oversubscribed mainly due to participants from business. The web provided new avenues for research not only for people in computer science but also in all areas of human learning. It has changed the way we do everything! Using simple words even a naive web user can find and subsequently access a large repository of web pages through the intermediary of a number of search engines. It is worthwhile to note at this point that the early search systems developed by the pioneers of the web have all but disappeared, replaced by late arrivals. The web, one of the services of the Internet, made it possible to create the vision that Vannevar Bush wrote about in 1945[83] in less than half a century!

Marshall McLuhan[65] noted that "The medium is the message" in relation to new media, namely radio and television, introduced in the early and mid-twentieth century. AM broadcasting was established in the 1920s and FM broadcasting in the 1940s. TV broadcasting started in a small way in the 1940s. With the advent of the web and its appendages, search engines, OSNs(on-line social networks) and the popularity of the mobile phone and its integration of the internet and web one wonders what the characteristic of it before looking at its contents. Most of these OSNs want to be THE internet and try to entice its users to be glued to them and never need to use anything but their system.

For a vast majority of users of the the internet, their principal or exclusive access point is a small screen with limited user interaction. Most web browsers, meant for this restricted media have very few user controls. With a limited visio-keyboard the interaction is awkward. The traditional menu at the top of a browser display is no longer a default and for some browsers it is impossible to access, even for more robust desktop versions of the applications.

Another major application of the web has been the introduction of OSNs, and other "platforms" to allow anyone to share personal information and news and to express their opinion on any topic[2]. Users seem to have no second thoughts in posting any type of personal information. Their lack of sophistication in assigning the correct setting for privacy to these postings means that their personal information may be accessed by anyone on the system and of course by the OSN that hosts this application[3]. The use of weak or "easy to guess" passwords do not help. The privacy issue has been of secondary importance for many of these OSN operators. These operators, through their terms of service, effectively take over perpetual ownership of all information and use it for commercial purposes and/or to sell to third parties. A privacy bill, recently proposed in the US Congress, would offer little help to individuals while giving companies great leeway in determining how they collect, use and share personal data[58].

## 4.2 Web and Artificial Intelligence

A proposal was made by McCarthy et. al. in August 1955[62] to set up a 20 man-month study in the summer of 1956 with the following goal:

"The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves." [62]

"It is generally agreed that this was the birth of AI. Recall that at that time the first generation of digital computers were around for less than a decade and were mammoths (in size and weight) with very little memory. More recently, a century long study of AI, to be hosted at Stanford University was published[80]. Over the past seventy-five years, there has been astronomical increases in computing power and storage capacity along with miniaturization; the progress in many aspects of computer science has also been remarkable. The computing power for a given volume and weight has increased by many powers of magnitude. This has enabled more complex algorithms, data storage and analysis to be possible. With the capitalization of the internet and the enormous potential for venture capitalists, the problem of funding was solved. This allowed the application and adaption of many approaches including half-baked concepts to be realized. The players included new companies supported to extend the commercialization of computing and the internet to new applications and the replacement of established ways of doing things by new ones. Many of these have destroyed or are destroying jobs and ways of life and have created tons of discarded devices replaced by new ones with different bells and whistles. Very little intelligence, artificial or natural, is required to see the adverse environmental impact of this madness."[20, 21, 28]

Artificial intelligence in the last two decades since the advent of the web has focused on the development of intelligent agents using reasoning based on statistics collected from a learning database. With the availability of extensive databases and computing capabilities, impressive progress has been made in speech recognition, image classification, machine translation, locomotion, and query-response systems. Most of these are already on-line and used by millions. One popular application is the live driving directions used by many drivers and which has made obsolete the good old map and preplanning of trips. It should be noted that the directions given are many times not the most efficient or least polluting ones: even though they may take the least time, the directions are not the shortest and environmentally friendly.

Most machine learning algorithms learn using programmed rules; whether it be simple or more complex neural networks. Usually, the learning algorithm and the programs for it take into account all the possible connections in the sample data and additional data that is generated by the learning. The problem is that there is always the possibility of missing some connections and of course the bugs introduced in the programming. Hackers have used the bugs and trap doors to create spyware.

Recently a tech-giant fired an engineer who claimed that the software system LaMDA (Language Model for Dialog Applications) is sentient. This was his conclusion after testing the system and according to the engineer the system displayed signs of experiencing sensation or feeling[45, 61]. According to the tech-giant the system is designed to generate convincing human language. One wonders about its use in customer support to give an impression to clients that they are talking to a human and hence needing fewer support staff.

The concept of conscience (intelligence) in a machine was conjectured by Samuel Butler, in his book Erewhon[4][12].

[2] These users have only to become familiar with this OSN interface and stay ignorant of the mechanisms used much less starting a text editor program such as EMACS, and type out simple compact web page.

[3] The terms of service that the user must have agreed to at the time of signing up, would make sure that the OSN is protecting itself and is able to mine and exploit the user's data.

[4] Published in 1872 and digitized in 2005 by the Gutenberg project

"There is no security against the ultimate development of mechanical consciousness, in the fact of machines possessing little consciousness now. A mollusc has not much consciousness. Reflect upon the extraordinary advance which machines have made during the last few hundred years, and note how slowly the animal and vegetable kingdoms are advancing. The more highly organized machines are creatures not so much of yesterday, as of the last five minutes, so to speak, in comparison with past time. Assume for the sake of argument that conscious beings have existed for some twenty million years: see what strides machines have made in the last thousand! May not the world last twenty million years longer? If so, what will they not in the end become? Is it not safer to nip the mischief in the bud and to forbid them further progress?"

However, this progress is going on. Businesses such as banks, utilities etc. are downloading most of the billing and payment operations to the internet or its surrogate mobile phone. They are saving all the mailing costs etc. and the cost of the internet/mobile device, plans and bandwidth is to be borne by the customers. None of the savings is passed to the customer. It is no wonder that these businesses are looking for a sentient system to replace what ever customer support they are providing and the big-tech that arrives there is going to reap the benefit. Companies are increasingly turning to chat-bots to interact with customers. The author had a bad experience with these bots and not being satisfied tried all means to get the system to get a human to interact without success. Companies are trying to make these bots more' human'. Is this what seems to be behind the recent story about a bot being, according to one engineer, sentient? These companies are developing better bots to interact with customers and hence cut their expenses[9].

A system such as LaMDA, if it could interact like a human[5] with customers would be a boon. Customer service provided by real employees, face-to-face, has been replaced by phone calls[6]; the first is disappearing and even the second is being replaced by a repeated message to send the organization an email which would be answered in 24 hours. All this transition to save money and not to hire telephone receptionists.

Looking at the case of LaMDA, one wonders if this is the intent to provide human like robot service to customers. Victor Frankenstein created a human beast which started learning from observations and became sentient[78]. He asked Frankenstein to create a companion which Victor resisted and for which he paid by his own life. It is likely that LaMDA and its clones will continue its evolution.

The big tech companies behind these bots could replicate these systems and could easily adapt the bots for other applications. From what one observes, it does not seem they, in the pursuit of profit and market domination, have the same reluctance as Victor Frankenstein who did not create a companion for the fiend he created[78]! It is likely that the internet will continue to metastasize into bot-ruled systems that will mimic humane nature. Companies would outsource customer service to big-techs instead of the developing countries since it would likely be cheaper and garner another big bonus for the CEOs and VPs IT!

Once bots take over these jobs handled by humans, there would be more profit for the business and fewer positions for the increasing population which is to reach eight billion souls by November 14, 2020i. Anders [3] had anticipated this situation on an "overmanned" world, wondering what will become of these billions of superfluous humans.

However, looking at the number of bugs in most systems and the recent incident in Canada when an entire communication network that serves millions of users went down during an update which likely had bugs,[57] one would have to expect disasters like this with a global foot print!

## 5 DIGITAL ECONOMY HUBRIS AND GREED IN CORRUPT SYSTEMS

As mentioned above, the development in the internet and connectivity has led to organization abandoning their computer and IT service for having all their data hosted on the cloud and the IT is supplied by software houses. The Government of Canada made the move to a system called Phoenix and has suffered the consequences. Universities have now taken up this initiative under the impression that there would be savings. The author's experiences with these systems have not been positive.

The chances of hacking of such systems are enormous with all the built in bugs and back-doors in all software. A recent example of this, reported in[79], occurred in the student-tracking software system that affected the confidential information of more than a million current and former school children. It appears that safeguards are not in place: one of the first things that should be considered when selecting a system seems to be missing.

One wonders if sleek marketing techniques had been used to sell such systems to eager VP-IT who wanted to take credit for the 'apparent' savings. We know how the tobacco industry had hooked millions of people on tobacco - a difficult dependency to overcome[37]. The opioid crisis is another example of the pharmaceutical companies, using marketing directly to medical doctors - the prescribers of drugs, and into the bodies of suffering patients and hooking them on an addictive pain relief drug[51]: yet another example of 'break things". When the patients, start dying, the same marketing and/or management consultants step in to repair their image and try to convert the evil into an opportunity. Behind the scenes, marketing and consulting organizations have guided the Opioid Crisis[48]. The investigation of tens of thousands of documents illustrates the working of consultants for opioid makers. Such firms become a trusted adviser to companies manufacturing and aggressively marketing opioids which is considered to be the cause of hundreds of thousands of lives. Such management consultants, like those for big tech, helped big pharma to develop a strategy for dealing with regulation bureaucracy to seek approval for products.

European Union authorities have been urged to investigate a former politician linked to Uber and consider stripping the cab-hailing company of access passes to the European parliament, amid growing calls to rein in tech lobbyists.[74]. The demand for an EU inquiry comes as some politicians consider tighter rules on lobbying after the publication of the Uber files, a trove of data

[5] Is it not the objective of big-tech?

[6] Which ends up being put on hold and subjected to inane messages.

leaked to the Guardian and shared with media in 29 countries via the International Consortium of Investigative Journalists.

Big-tech and many other businesses, using lobbying, influence peddling and the misplaced belief that the high level leaders of innovation must not be stifled with regulations have been able to ignore existing regulations, laws and practices. Their modus operandi seems to be to break things(ignore everything - and then fix things they have broken to their own advantage). This includes the time honoured copyright laws etc[7]. Some of the machinations used by some companies have been revealed by what is being called the Facebook papers and the Uber papers.

The emergence of the web as an application of the internet ushered in the fourth (or the fifth) industrial revolution. As in the very first revolution, it changed many aspects of life as new corporations were set up using venture capitals and they were able to ignore all tradition, rules and regulations using lobbying and bent politicians to get the regulations etc. changed!

The philosophy used is that of ignoring all norms, traditions, regulations and laws. By getting bent politicians and their aides on-side they either have these not applied to them or have the in-place regulations and laws changed. The many bent politicians[8] would oblige these big-tech companies! Some of them, when interviewed by the media about the Uber papers, seem to be proud of their accomplishment in this connection. They and these new tech companies can use their fortunes to hire lawyers and use the courts to defer legal recourse. A case in point is the case of a Canadian woman, Deborah Douez, who has been battling one of the big tech companies now approaching a decade[14]. The bent politicians also use this very tool to raise millions of dollars from ignorant people, having nothing better to do than follow people like them and be taken in by their lies end up sending in contributions to whatever cause. Recently one of these is reported to have raised over 250, 000, 000 USD. The funds are augmented by other billionaires, some of them have been using the internet to mint a fortune.

An organization which was supposed to promote the hypermedia protocol was taken over by business. One is appalled by the sophisticated tracking incorporated in the browsers and the applications for mobile devices, all of which have access to the data on the phone instead of requesting data if and when required and getting the permission of the user for any bit of information. The design of these systems allows these breaches in the user's privacy.

The Digital Millennium Copyright Act (DMCA), signed into law in 1998, provided complete immunity to internet service providers and platforms from copyright claims when their users upload or share copyrighted material to the platform. Thus the law clears the the platform from immediate liability and it is likely that the material would stay for a considerable length of time on the platform[52].

The big tech companies [59] have big purses to hire lobbyists, finance the politicians' election campaign, use lawyers to delay and fight one and every one[10]. Their only concern is to cannibalize and monopolize and at the same time colonize using the US government to shield them and lobby the foreign governments[85]. For this they either put any competitor out of business or buy up any start-up and competition[95–97, 99]. It is likely that many of the people at the acquired company may become redundant! The successive Usain governments have not blocked any such buyouts as had happened with the long list given on the Wikipedia pages mentioned above. There seems to be easy access to decision-makers at all level of governments by big techs: this access has been used to influence the decison makers and most of them are happy to be photographed with the CEOs of these big-techs. These corrupt leaders believe that these tech giants were providing growth and innovation while in reality they were stifling competition and destroying existing infra-structure and competition which provides a choice to the consumers.

The US system has abandoned a Global Tax act, which was aimed at cracking down on companies evading taxes by shifting jobs and profits around the world and the US system is failing to raise tax rates on these multinational corporations[72]. For example, some of these big-tech companies book its profits in one country as being made in another one to minimize its exposure to corporate taxation. Most taxation agencies aggressively go after individuals and local small businesses, while ignoring major big tech companies[49].

Some of the business models used by so many Internet based companies use the "relies on privatizing profit and socializing risks"[18] and exploitation. of one kind or another. They are led by ruthless people who seem to have an unquenchable appetite to monopolize not only their segment of the business, but looking for opportunities to expand their horizons. There are many self-serving politicians, policy makers and their aides who see personal gain. They use their connections and prestige to continue to influence the government even after their term of office end. One needs to look at some of the opulent properties some of the ex-leaders have acquired!

Not long ago, we may have believed that technology would enhance personal freedom and democratic choice. It looked to be so for a while! However, technology is starting to shift the global balance toward monopolies and autocratic regimes. Furthermore conflicts between democracies and autocracies have already started[4]. As reported by Amnesty International, the business model of the OSNs is threatening human rights[5].

## 6 EXPLOITATION OF THE INFORMATION AGE

The introduction of web made the internet accessible to lots more people along with the rapid use of mobile phones integrated seamlessly to the internet. This was also the start of the spread of mis-information! The mis-information is such that it triggers emotions in the recipient, who in turn propagates this information, directly or automatically thanks to the algorithms used by the OSN! Instead of the media being the message, the aroused users become the messengers to others and hence reinforce the mis-information.

As pointed out by Gunther Anders[3], as in the previous industrial revolutions, big tech companies of the information age consistently oppose unionization efforts and use the old technique of finding jurisdictions with more favourable lax regulations, looser workplace requirements, and almost no consequences for breaking labour laws[44, 75].

The workings of these big tech companies are coming to light thanks to thousands of papers leaked by insiders from some of these digital economy based conglomerates: it is expected more would be

[7] Perhaps developing countries should ignore all patents, and copy algorithms while improving them!

[8] They used these OSN to propel themselves to their job and want to continue to keep climbing!

forthcoming in the future for others of these companies in the digital economy. What we find is that the OSNs and the big-techs consider the user data, actions etc., as a mine to be appropriated, exploited and bring to light anything that is concealed or connected[3].

The trove of documents released by an ex-employee of Facebook reveal, among other things, the role of the company in the Jan. 6 insurrection in Washington, D.C. and the effect of the company around the world. While privately and meticulously tracking the hate and divisiveness magnified by this OSN platform, it has not heeded warnings from its engineers about the dangers posed by the design decisions made for its algorithms with the goal of having users stay riveted to and interacting with the site; the OSN chooses growth through maximum engagement over user safety. The public claims made by this OSN often conflict with internal research. One of these is the claim of removing 95 percent of hate speech when in reality it is only 5 percent (or did they get the figures mixed up)[8, 55].

The OSN's problems with hate speech and misinformation are dramatically worse in the developing world. Due to weaker moderation in many countries OSNs allow their platforms to be used by maleficent actors and authoritarian regimes to propagate hateful and divisive mis-information. The head of Facebook, who controls the majority of the voting shares of the company, told Congress that it was "not at all clear" that social networks polarize people, when Facebook's own researchers had repeatedly found that they do[34]!

In an experiment in 2019, a pair of Facebook's employees set up a dummy account for a 21 year old woman in India, the company's largest market. Without any input from this dummy account, the feed to it was first filled with pornography and, soon after, it was flooded with propaganda favourable for the then prime minster[9] and anti-minority hate speech. One reason could be the "reward" changes made in this OSN algorithms. It appears that while the OSN pushed into the developing world it didn't invest in protections anywhere near the ones in the US context, themselves woefully inadequate[63, 101].

While the program called 'free basic' was initially attempted by Facebook in India, opposition forced this OSN to abandon it[21]. However it appears that Facebook did not give up this ploy and has pushed the 'free basic" program in other countries e.g., Ghana, Mexico and Myanmar. Facebook has been able to push this program allowing people there to experience this OSN to be the internet tout court. As in their failed attempt in India, Facebook partnered with local telecom operators in these countries to give free access to its own platform along with a bundle of other basic services like job listings and weather reports. This scheme has locked millions of people into a version of the Internet controlled by this single OSN [101].

In many OSN applications users are the messengers since they can forward messages to their friends and groups without checking the authenticity of the message[33]. By having users to spread the messages on many of the OSN, these companies are using free labour: in addition to having the users invite their friends and family to be part of the platform, this creates a positive feedback loop. One may understand the small mom-and-pop store using the hosting facility of the OSNs however, one wonders at the mental savvy of managers of many large public institutes, such as universities, which allow themselves to be part of these platforms by displaying the logos of these OSN on their home pages and using the OSN 'free' hosting facility.

There are essentially two operating systems for most of the computers in the world and just two operating systems for mobile phones. These OSs are derived from a version of the very same open source operating system, Linux: which is, in tuen, based on Unix, an older operating system. However, after years and years of development and many releases, they are full of bugs, loopholes and trap doors. The case of the journalist who was killed in an embassy in Turkey is well known - one of the factors which led up to this was a commercial spyware from a company in the Middle East which allowed this journalist and others to be spied on through their mobile phones[54, 86]. Was the spyware possible due to some vulnerability in the the mobile operating system and has it been fixed? How easy it is to do something similar with these weak systems is further illustrated by the case reported recently of a 15 year old hacker who was able to break into the mobile system to create hacking spyware. This spyware was sold to tens of thousands of domestic violence perpetrators[67].

Intrusion into the life of users and constant surveillance due to the buggy nature of the mobile phone operating systems is made by a corporation which terminate employees for giving their opinion to the world. In the case of a coffee franchise, the application tracked the user: most of the data was collected even when such an application was not being used. This is an intrusion and a loss of users' privacy. Hardly any user, knowingly, downloads and installs such applications for the benefit of the business which offers the application nor any third parties. In exchange for one's privacy, the meagre compensation offered by this chain of coffee purveyors is a cup of coffee and a donuts [7, 45, 100].

After last year's whistle-blowing revelations relating to practices at Facebook, the Uber files published, recently by The Guardian, constitute another seminal big tech morality tale[36]. The vast cache of documents –leaked by a former key public relations employee – offers an insight into a digital giant as it sought to expand at any cost. At the same time, it chronicles the complicity of a political class which, itself drunk on big-tech concoctions, went along for the ride.

The so called ride sharing service was introduced to provide in effect a taxi-service without having to own a fleet of taxis, get permits for them from the local municipal authority, pay auto-insurance or hire drivers to drive them. The fact that the livelihood of tens of thousands of taxi drivers and their licenses, which could have cost thousands of dollars would be devalued is completely ignored not only by this company but also by the politicians who championed this scheme. The scheme was to have individual drivers with cars to provide a taxi-service using an application on a mobile phone. The way this business was set up is outlined in thousands of documents from the company released by a lobbyist who was associated with the company: these are called the Uber files[35]. According to a Guardian editorial this trove of documents "offers a unique and salutary insight into the arrogance and hubris of a digital giant"[81]. The approach used by the company, consistent with those of the big techs, was to infringe on existing regulations,

[9] Modi running for re-election

show the politicians the benefits of the company for their self-interest, and have them change the laws or make the laws not applicable to them. In the case of Uber, these politicians did not recognize that the local and national taxi service could be given an opportunity to devise such a software system and provide local jobs to software developers and use idle computing from local data centres.

Uber also used a a tool called Greyball, to identify officials acting as the companies client by using the data collected from their own application and others to avoid being detected in many cities and countries[50]. This tool was used to dominate this fake taxi-business. Uber also treated its drivers as contractors, not providing any benefits and by showing them goals that they could achieve which pushed them to work longer. They gave away large grants to academics to provide them with strategy of feed to the media[40]. Looking at the Uber paper, raises a question: Did the politicians take orders from the Uber executives[35]?

Some early enthusiasts of OSNs are finally waking up and uttering 'mea culpa" as they realize the evil that is being done by the meta-stised internet today. The intimacy of the Big-tech companies with the Usain leaders was one of the reasons that the mergers were allowed to happen: "Obama's regulators allowed Facebook to buy up its biggest competitors — first Instagram, then WhatsApp — and failed to crack down on its recklessness with users' private data"[60]. These same leaders used the OSNs reach to get elected, re-elected and collect funds. The fundraising bit is evidenced by the amount collected by unsavoury players who sacrifice principles, reality and their oaths for self-service and imagined wrongs[45, 64].

The metastasis of the internet has allowed a handful of individuals, exploiting the groundwork done by academicians and researchers to transform systems that were supposed to allow connectivity, in such way that is detrimental to all societies and human rights including privacy and democracy.

## 7 NEED FOR A NEW BEGINNING

Humans need a better web, better IOTs, better mobile devices, better software, better protection against monopolies, and of course better politicians and systems of government. Unfortunately many revolutions did not improve the lot of the ordinary person.

The internet, through the web application and mobile systems, has revolutionized the world with over 4 billion people using this media. They read news, send emails and text messages are able to have video conversations and find answers to questions. Yet when these billions participate in online-life most of them rely heavily on the services of just two corporations who control operating systems for the mobile devices. The mobile devices are used to access the services so integral that it is difficult for them to use the internet without these devices

One of the rays of hope is steps taken by the European Union[76]. Proposed EU legislation would force internet services to combat misinformation and publicize their roles amplifying divisive content and stop targeting ads based on ethnicity, religion or sexual orientation. The law is an attempt to address OSN's harm and requiring them to be more pro-active in monitoring their platform for illicit content or risk billions of dollars in fines. Tech companies would be compelled to set up new policies and procedures to rapidly remove flagged hate speech, terrorist propaganda and other material defined as illegal by countries within the European Union. This law is putting an end to self-regulation which had not previously been done since growth was put above monitoring the contents!

Laws such as the above need to be passed in other parts of the world. However what is more important is to take this time to provide a communication system as a necessary utility for all by the public authorities to complement the postal system. Failures such as the recent one in the Canadian communication system offered by a private, for profit, organization should become a cautionary tale [57].

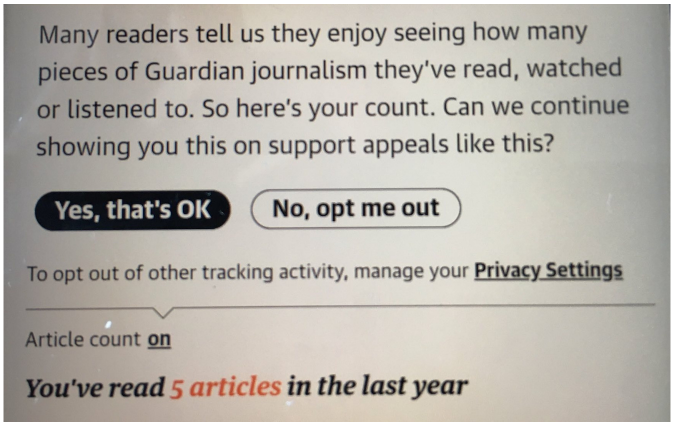
Currently, there are just a limited number of for-profit US-based corporations, which offer web search, email, mobile and other operating systems. Software has moved from being 'sold' to licensed to provide a steady stream of income for these companies. The existence of open source software is most likely being used as a shield to protect these behemoths from being treated as monopolies. The unfortunate thing is that many people, even IT professionals, do not use open source software!

As pointed out in [1, 29], there is an urgent need to set up a global Software Assurance Agency to certify all software regardless of its origin. The concept is similar to CSA[17], UL[82]. All software has to go through a certification by this agency.

Countries should align to put an end to having billions of mobile devices managed by operating systems controlled by just two companies. However, we have to address the duopoly of software - given that this is how people access large parts of contemporary reality, this software must be liberalized and controlled by a public agency. Also open source software must be adopted in schools and universities. The trend of using one system that seems to be in vogue must end.

OSNs and their new algorithms should be required to pass some kind of stress test with their systems before their actual deployment: this should be monitored and certified by a global agency such as the proposed SAA. Since humans are sensitive to and respond to emotional triggers: they also share messages that reinforces their beliefs: hence, algorithms must be checked to prevent creating hatred and animosity[84].

The feeble attempts of some governments, e.g., the Canadian Bill C-18, to level the playing field by making the OSNs pay for the news they use which may be produced by struggling small and medium news organizations. The charade that things are "free" must end and a reasonable charge should be put on contents that are really not free. Allow each unit of content to carry a competitive micro-price-tag. This would also allow the removal of all paywalls from all news media sites. Since the end user is paying for the use of the internet connection and the amount of data used, some portion of the charges made by the ISPs should flow back to the original producers of the news based on the micro-price-tag. As illustrated below, most users consume only a fraction of the news put out by any one publication, the micro-charges would be reasonable and could easily be built into the fees charged by the ISP. By making news accessible from the original responsible source, people will spend more time following real news rather than be fed junk by the OSNs.



Many readers tell us they enjoy seeing how many pieces of Guardian journalism they’ve read, watched or listened to. So here’s your count. Can we continue showing you this on support appeals like this?

To opt out of other tracking activity, manage your **Privacy Settings**

Article count **on**

***You've read 5 articles in the last year***

**Figure 3: Number of articles read**

***A pay wall system for news media do not allow users to read the articles. In the case of some newspapers, the paywall keeps giving messages for supporting the media and keeps a count of articles read.***

It is the current practice that many large news organizations offer the digital content to a user for a modest amount such as one dollar a week. However, this is an exhorbitant amount considering the fact the most users read several news outlets. However, the amount of material used from each such outlet is relatively small. Hence, automatically transferring a portion of the internet connect monthly charge to the source of the information would solve the problem, without a multitude of digital subscriptions.

The above scheme, in addition to being simple, would allow all ads in the contents removed since the end user is paying for the contents. Eliminating the ads would save bandwidth and energy; the latter is good for the environment. This solution would mean that there would be no need for pay-walls and nagging requests to sign up or being put on a mailing list for headlines etc. Web browsers should not ever allow trackers and third party cookies to restore the web to its original spirit of sharing.

Considering the fact that the big-techs are using the ideas and even the software algorithms released by previous generation systems, and exploiting these openly accessible concepts to create systems that is exploiting the human race and amassing a fortune while the needs of millions of humans including children are not met. One looks at one of the ideas of property by Locke who considered a person’s work as his property as long as enough is left for the common good of others [56].

“Sec. 27. Though the earth, and all inferior creatures, be common to all men, yet every man has a property in his own person: this nobody has any right to but himself. The labour of his body, and the work of his hands, we may say, are properly his. Whatsoever then he removes out of the state that nature hath provided, and left it in, he hath mixed his labour with, and joined to it something that is his own, and thereby makes it his property. It being by him removed from the common state nature hath placed it in, it hath by this labour something annexed to it, that excludes the common right of other men: for this labour being the unquestionable property of the labourer, no man but he can have a right to what that is once joined to, at least where there is enough, and as good, left in common for others.”

**Figure 4: Copy Forward**

***The concept of CopyForward was put forward by the author to not let baron-enterpreneurs to exploit human knowledge for private gain. It depends on moral obligation with the hope that it will help the coming generations.***

There are various forms for ’protecting’ a person’s work which is their property: the usual is copyright. In the digital age, it is becoming difficult to enforce this! The author has come up with CopyForward, given below, which allows a digital content as the property of the person creating it but also to share in the sense that could be determined by the creator as given below:

“The document/work, in digital/electronic form, could be used for personal use and/or study, free of charge. Anyone could use it to derive updated versions. The derived version must be published under CopyForward. All authors of the version used to derive the new version must be included in the updated version in the existing order, followed by name(s) of author producing the derived work. Such derived version must be made available free of charge in electronic/digital form under CopyForward. Any other means of reproduction requires that part of the profit(income minus the actual production cost), not less than a third(33.33 percent), should be shared with established charitable organizations for children. Persons who found this document/work or any derived work useful are encouraged to also make a donation to the author(s) and/or their favourite charity.

Make sure to choose a charity which has very modest administrative charges(NOT more than 20% of their entire annual budget) or some deserving children in your own community.”

One of the first CopyForward items can be accessed from the Spectrum library[31].

Wonder if the people who released the software and the concept being used by the big-tech had used CopyForward!

In the meantime, the author hopes that the IT community would set up a project to realize the proposal[30] to allow an ordinary person to set up her own email and web server. The ownership of data could than be reclaimed and there would be little need of these big-techs.

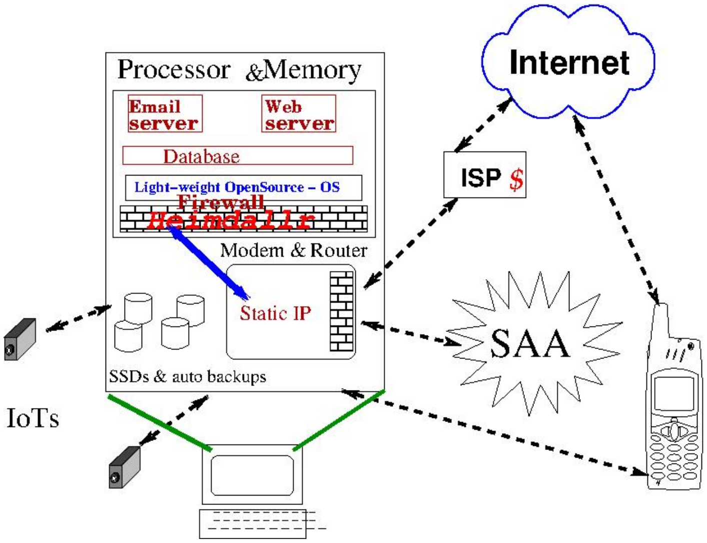


**Figure 5: Heimdallr**

***This is a system diagram to illustrate a turn-key system that could be marketed as a replacement for a modem-router - it has built in email and web server to allow anyone to reclaim the custody of her personal data.***

## REFERENCES

[1] Aksoy, Ayberk; Desai, Bipin C: *Heimdallr_1: A system design for the next generation of IoTs,* ICNSER2019: March 2019 pp 92–100 https://doi.org/10.1145/3333581.3333590

[2] AAI, America in the Age of Information, July 6-7, 1995, Lister Hill Center, Bethesda MD, http://users.encs.concordia.ca/ bcdesai/Age-of-Information-July-1995.pdf

[3] Anders, Günther *The obsolescence of man - Volume 2,* https://files.libcom.org/files/ObsolescenceofManVol II Gunther Anders.pdf

[4] Appathurai. James *Tech is enabling autocrats. Here's how to fight back* The Globe and Mail, 24 March, 2922 https://www.theglobeandmail.com/opinion/article-tech-is-enabling-autocrats-heres-how-to-fight-back/

[5] Amnesty International: *Surveillance Giants: How The Business Model Of Google And Facebook Threatens Human Rights,* 21 Nov, 2019, Index Number POL 30/1404/2019 https://www.amnesty.org/en/documents/pol30/1404/2019/en/

[6] Albertus, Michael; Menaldo, Victor: *Aftermath of Revolution*, NY Times, Feb. 14, 2013 https://www.nytimes.com/2013/02/15/opinion/global/aftermath-of-revolution.html

[7] Al Mallees, Nojoud: *App's data tracking resulted in loss of users' privacy, says report by federal, provincial authorities,* CBC News · Jun 01, 202 https://www.cbc.ca/news/business/tim-hortons-app-report-1.6473584

[8] Albergotti, Reed *FACEBOOK UNDER FIRE* Wahington Post, 26 Oct. 2021 newblock https://www.washingtonpost.com/technology/2021/10/26/frances-haugen-facebook-whistleblower-documents/

[9] Bogost, Ian *Googles Sentient Chatbot Is Our Self-Deceiving Future.* The Atlantic, June 2022 https://www.theatlantic.com/technology/archive/2022/06/google-engineer-sentient-ai-chatbot/661273/

[10] Brittain, Blake: *Meta hit with trademark lawsuit over new infinity-symbol logo*, Reuters, May 2, 2022 https://www.reuters.com/legal/litigation/meta-hit-with-trademark-lawsuit-over-new-infinity-symbol-logo-2022-05-02/

[11] Behr, Rafael *The Uber files tell a simple truth: democracy depends on curbing mercenary tech giants,* The Guardian, https://www.theguardian.com/commentisfree/2022/jul/11/uber-files-democracy-silicon-valley

[12] Butler, Samule *Erewhon or Over the Range* Original 9 June, 1872. Prjoject Gutenberg March 20, 2005 https://www.gutenberg.org/files/1906/1906-h/1906-h.htm

[13] *Charles Babbage*, Wikipedia https://en.wikipedia.org/wiki/Charles_Babbage

[14] CBC, The Canadian Press *B.C. court allows class-action lawsuit against Facebook to expand* The Canadian Press, 14 May, 2019 https://www.cbc.ca/news/canada/british-columbia/facebook-class-action-expansion-1.5135031

[15] Chen, Brian X.: *I Downloaded the Information That Facebook Has on Me. Yikes,* NY Times, Apr. 11, 2018, https://www.nytimes.com/2018/04/11/technology/personaltech/i-downloaded-the-information-that-facebook-has-on-me-yikes.html

[16] Charette, Robert N.: *Canadian Government's Phoenix Pay System an "Incomprehensible Failure": That's the nicest thing that could be said for a debacle of the first rank,* IEEE Spectrum, 05 Jun 2018 https://spectrum.ieee.org/riskfactor/computing/software/ canadian-governments-phoenix-pay-system-an-incomprehensible-failure

[17] *CSA Group* https://www.csagroup.org/

[18] Cann, Vicky *Uber's privileged access to politicians shows the lobby system urgently needs to change,* The Guardain, Mon 11 Jul 2022 https://www.theguardian.com/commentisfree/2022/jul/11/uber-privileged-access-eu-politicians-lobby-system-change

[19] Desai, Bipin C. *Technological Singularities*, IDEAS 15, July 13 - 15 2015, Yokohama, Japan http://dx.doi.org/10.1145/2790755.2790769

[20] Desai, Bipin C. *The Web of Betrayals*, IDEAS 2018, June 18–20, 2018, Villa San Giovanni, Italy https://doi.org/10.1145/3216122.3216140

[21] Desai, Bipin C.: *Colonization of the Internet*, IDEAS 2021, July 2021, pp 36–45 https://doi.org/10.1145/3472163.3472179

[22] Desai, Bipin C.: newblock *Report of the Navigation Issues Workshop*, Computer Networks and ISDN Systems, Vol. 27-2, November 1994, pp. 332-333.

[23] Desai, Bipin C.; Swiercz, Stan: *WebJourney: Visualization of a Web Journey,* In: Digital libraries: research and technology advances: selected papers: ADL'95 Forum, McLean, Virginia, USA, May 15-17, 1995. Lecture notes in computer science (1082). Springer, Berlin, pp. 63-80. ISBN 9783540614104, https://spectrum.library.concordia.ca/983869/

[24] Desai, Bipin C.: *Test: Internet Indexing Systems vs List of Known URLs*, June, 1995, available on the Web from https://users.encs.concordia.ca/∼bcdesai/test-of-index-systems.html, https://spectrum.library.concordia.ca/983875/

[25] Desai, Bipin C.: *Test: Internet Indexing Systems vs List of Known URLs: Re visited,* October 1997, available on the Web from https://users.encs.concordia.ca/∼bcdesai/test-of-index-systems-revisited.html, https://spectrum.library.concordia.ca/983876/

[26] Desai, Bipin C.; Pinkerton, Brian (ed): *Proceedings of the WWW III Workshop on Web-wide Indexing/Semantic Header or Cover Page,* Darmstadt, Germany, April 1995, https://users.encs.concordia.ca/∼bcdesai/www3-wrkA/www3-wrkA-Proc.ps.gz

[27] Desai, Bipin C.: *Search and Discovery on the Web,* October 2001, https://spectrum.library.concordia.ca/983874/

[28] Desai, Bipin C.: *The state of data*, IDEAS2014, Portugal July 2014, 77-86, ISBN: 978-1-4503-2627-8 DOI 10.1145/2628194.2628229

[29] Desai, Bipin C.: *IoT-Imminent Ownership Treat,*, Proc. IDEAS 2017, Bristol, July 2017, pp 82-89, DOI 10.1145/3105831.3105843

[30] Desai, Bipin C.: *Privacy in the Age Of Information (and algorithms)* IDEAS 2019, June 2019, Athens, Greece https://doi.org/10.475/3331076.3331089

[31] Desai, Bipin C.;Kipling, Arlin L *Database Web Programming,* BytePress, 2020, ISBN 9781988392066 9781988392042 https://spectrum.library.concordia.ca/id/eprint/988529/2/WebDB-Desai-Kipling-Oct-2020.pdf

[32] Desai, Bipin C.: *An Introduction to Database Systems,* West, St. Paul, MN. 1990, ISBN 0-314-66771-7, https://spectrum.library.concordia.ca/id/eprint/988586/1/An-Introduction-to-Database-Systems-Bipin-C.DESAI.pdf

[33] Dwoskin, Elizabeth; Owen, Annie G : *On WhatsApp, fake news is fast — and can be fatal,* Washington Post, 23 July, 2018 https://www.washingtonpost.com/business/economy/on-whatsapp-fake-news-is-fast–and-can-be-fatal/2018/07/23/a2dd7112-8ebf-11e8-bcd5-9d911c784c38_story.html

[34] Dwoskin, Elizabeth; Newmyer, Tory; Mahtani, Shibani: *The case against Mark Zuckerberg: Insiders say Facebook's CEO chose growth over safety* Wahington Post, 25 October, 2021 https://www.washingtonpost.com/technology/2021/10/25/mark-zuckerberg-facebook-whistleblower/

[35] Davies, Harry; Goodley, Simon; Lawrence, Felicity; Lewis, Paul; O'Carroll, Lisa: *Uber broke laws, duped police and secretly lobbied governments, leak reveals,* The

Guardian, July 11. 2022 https://www.theguardian.com/news/2022/jul/10/uber-files-leak-reveals-global-lobbying-campaign

[36] Davies, Harry; Goodley, Simon; Lawrence, Felicity; Lewis, Paul; O'Carroll, Lisa: *The Uber files* The Guardian, 11 Jul 2022 https://www.theguardian.com/news/series/uber-files

[37] Dani, John A.; Balfour, David J.K. *Historical and current perspective on tobacco use and nicotine addiction* Review Historical Perspective|, 14-7, P383-392, July 01, 2011 https://doi.org/10.1016/j.tins.2011.05.001

[38] Davies, Rob; Goodley, Simon *Uber bosses told staff to use 'kill switch' during raids to stop police seeing data* The Guardain, 10 July, 2022 https://www.theguardian.com/news/2022/jul/10/uber-bosses-told-staff-use-kill-switch-raids-stop-police-seeing-data

[39] Favreau, François-Alexis *Sonneur d'alerte chez Uber: le lobbyiste Mark MacGann se dévoile*, SRC , 11 July 2022 https://ici.radio-canada.ca/nouvelle/1897240/lanceur-alerte-uber-mark-macgann-systeme-mensonge?

[40] Lawrence, Felicity *Uber paid academics six-figure sums for research to feed to the media* The Guardian, 12 July, 2022 https://www.theguardian.com/news/2022/jul/12/uber-paid-academics-six-figure-sums-for-research-to-feed-to-the-media

[41] Guinan, Joe; O'Neill, Martin: *Only bold state intervention will save us from a future owned by corporate giants*, The Guardian, 6 Jul 2020, https://www.theguardian.com/commentisfree/2020/jul/06/state-intervention-amazon-recovery-covid-19

[42] Greenberg, Martin *The Computers of Tomorrow*, The Atlantic, 1964 https://www.theatlantic.com/magazine/archive/1964/05/the-computers-of-tomorrow/658239/

[43] Gibbs, Samuel: *Apple fixes HomeKit bug that allowed remote unlocking of users' doors* , The Guardian 8 Dec. 2017 https://www.theguardian.com/technology/2017/dec/08/apple-fixes-homekit-bug-remote-unlocking-doors-security-flaw-iphone-ipad-ios-112-smart-lock-home

[44] Greenhouse, Steven: *Amazon chews through the average worker in eight months. They need a union*, The Guardian, Feb 4, 2022, https://www.theguardian.com/commentisfree/2022/feb/04/amazon-chews-through-the-average-worker-in-eight-months-they-need-a-union

[45] Goldmacher, Shane; Haberman, Maggie: *Trump Raises $170 Million as He Denies His Loss and Eyes the Future* The NY Times, Aug. 7, 2021 https://www.nytimes.com/2020/11/30/us/politics/trump-campaign-donations.html

[46] Guardain Staff *Google fires software engineer who claims AI chatbot is sentient*, The Guardain staff, 23 Jul 2022 https://www.theguardian.com/technology/2022/jul/23/google-fires-software-engineer-who-claims-ai-chatbot-is-sentient

[47] Hern, Alex *TechScape: suspicious of TikTok? You're not alone*, The Guardian, 0 Jul 2022 https://www.theguardian.com/technology/2022/jul/20/tiktoks-privacy-problem-isnt-what-you-think

[48] Hamby, Chris; Forsythe, Michael: *Behind the Scenes, McKinsey Guided Companies at the Center of the Opioid Crisis* New York Times, June 29, 2022 https://www.nytimes.com/2022/06/29/business/mckinsey-opioid-crisis-opana.html

[49] Hager, Mike; O'Kanet, Josh: *Business groups, Tories seek changes to Canadian tax system after Amazon findings* GLobe and Mail, July 18, 2022 https://www.theglobeandmail.com/business/article-industry-groups-opposition-call-for-changes-to-canadian-tax-system/

[50] Isaac, Mike *How Uber Deceives the Authorities Worldwide* New York Times, 3 March, 2017 https://www.nytimes.com/2017/03/03/technology/uber-greyball-program-evade-authorities.html

[51] Keefe, Patrick Radden: *Empire of Pain* Doubleday Books, 2021 ISBN 13:9780385545686

[52] Knox, Ron *The copyright kille* 11 January 2019

[53] Kermani, Secunder: *Pakistan activists targeted in Facebook attacks*, BBC, May 15, 2018, http://www.bbc.com/news/world-asia-44107381

[54] Kirchgaessner, Stephanie *Saudis behind NSO spyware attack on Jamal Khashoggi's family, leak suggests* The Guardian, 18 Jul 2021 https://www.theguardian.com/world/2021/jul/18/nso-spyware-used-to-target-family-of-jamal-khashoggi-leaked-data-shows-saudis-pegasus

[55] Lima, Cristiano *A whistleblower's power: Key takeaways from the Facebook Papers* Washington Post, 26October, 2021 https://www.washingtonpost.com/technology/2021/10/25/what-are-the-facebook-papers/

[56] Locke, John *Second Treatise of Government* 1688, Gutenberg 2005 https://www.gutenberg.org/ebooks/7370

[57] Logan, Nick· *Rogers network outage a 'wake-up call' after business, essential services disrupted* CBC News 8 July, 2022. https://www.cbc.ca/news/business/rogers-outage-no-plan-b-1.6515664

[58] McCabe, David: *Congress and Trump Agreed They Want a National Privacy Law. It Is Nowhere in Sight.* NYTimes Oct. 1, 2019 https://www.nytimes.com/2019/10/01/technology/national-privacy-law.html

[59] Manjoo, Farhad *Tech's 'Frightful 5' Will Dominate Digital Life for Foreseeable Future Give this article* NY Times 20 Jan., 2016 https://www.nytimes.com/2016/01/21/technology/techs-frightful-5-will-dominate-digital-life-for-foreseeable-future.html

[60] Manjoo, Farhad *I Was Wrong About Facebook* NY Times. 21 July, 2022 https://www.nytimes.com/2022/07/21/opinion/farhad-manjoo-facebook.html

[61] McQuillan, Laura: *A Google engineer says AI has become sentient. What does that actually mean?*, CBC, 24 Jun, 2022 https://www.cbc.ca/news/science/ai-consciousness-how-to-recognize-1.6498068

[62] McCarthy. J; Minsky, M. L.; Rochester. N; Shannon, C.E.: Bell Telephone Laboratories *A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE* 31 August, 1955 https://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html

[63] Merrill, Jeremy B. ; Oremus, Will: *Five points for anger, one for a 'like': How Facebook's formula fostered rage and misinformation* Washington Post, 26 October, 2021 https://www.washingtonpost.com/technology/2021/10/26/facebook-angry-emoji-algorithm/

[64] McEvoy, Jemima *Trump Raised $250 Million Since Election To Challenge Outcome—Here's Where Most Of The Money Will Actually Go* Forbes, 31 Jan., 2021, https://www.forbes.com/sites/jemimamcevoy/2021/01/31/trump-raised-250-million-since-election-to-challenge-outcome-heres-where-most-of-the-money-will-actually-go

[65] McLuhan, Marshall: *Understanding Media: The Extensions of Man*, McGraw-Hill. 1964

[66] Marshall McLuhan, Bruce R. Powers, *The Global Village: Transformations in World Life and Media in the 21st Century*, Oxford University Press, 1992.

[67] McGowan, Michael *Brisbane teenager built spyware used by domestic violence perpetrators across world, police allege* The Guardain, 30 July, 2022 https://www.theguardian.com/australia-news/2022/jul/30/brisbane-teenager-built-spyware-used-by-domestic-violence-perpetrators-across-world-police-allege

[68] Niiler, Eric *How the Second Industrial Revolution Changed AmericansĹives* https://www.history.com/news/second-industrial-revolution-advances

[69] Peter, Ian: *The history of email*, http://www.nethistory.info/History of the Internet/email.html

[70] Peter, Ian: *History of the World Wide Web*, http://www.nethistory.info/History of the Internet/web.html

[71] Paul, Kari; Milmo, Dan: *Mark Zuckerberg to face deposition over Cambridge Analytica scandal*, The Guardian, 20 Jul 2022, https://www.theguardian.com/technology/2022/jul/20/mark-zuckerberg-deposition-cambridge-analytica-facebook

[72] Rappeport, Alan; Tankersley, Jim: *How Joe Manchin Left a Global Tax Deal in Limbo T reasury Secretary*, New York TImes July 18, 2022, https://www.nytimes.com/2022/07/18/us/politics/joe-manchin-tax.html

[73] Riekeles, Georg: *I saw first-hand how US tech giants seduced the EU – and undermined democracy*, The Guardian, 28 June 2022, https://www.theguardian.com/commentisfree/2022/jun/28/i-saw-first-hand-tech-giants-seduced-eu-google-meta

[74] Rankin Jennifer: *EU urged to investigate ex-politician's Uber links and rein in tech lobbyist* The Guardian, 12-Jul, 2022, https://www.theguardian.com/news/2022/jul/12/eu-urged-investigate-ex-politician-uber-links-rein-in-tech-lobbyist

[75] Samaha, Albert: *How Amazon Exported American Working Conditions To Europe*, Buzzfeed, June 23, 2022, https://www.buzzfeednews.com/article/albertsamaha/amazon-poland-slovakia-czechia-germany-labor-laws

[76] Satariano, Adam: *E.U. Takes Aim at Social Media's Harms With Landmark New Law* The New York Times, 22 April, 2022, https://www.nytimes.com/2022/04/22/technology/european-union-social-media-law.html

[77] Smith, Justin E. H.: *The Internet Is Not What You Think It Is: A History, a Philosophy, a Warning*, Princeton University Press, 2021, ISBN 13: 9780691229683

[78] Shelley, Mary W.: *Frankenstein or, The Modern Prometheus*, Henry Colburn and Richard Bentley, 1831, Gutenberg, 2013 , https://www.gutenberg.org/ebooks/42324

[79] Singer, Natasha: *A Cyberattack Illuminates the Shaky State of Student Privacy*, NY TImes, 31 July, 2022, https://www.nytimes.com/2022/07/31/business/student-privacy-illuminate-hack.html

[80] 2021 Study Panel Report: *Gathering Strength, Gathering Storms*, Standford Univeriry, Sept, 2021, https://ai100.stanford.edu

[81] Editorial: *Devastating leaked documents underline the pressing need for proper regulation of the digital economy*, The Guardian 11 Jul 2022 , https://www.theguardian.com/commentisfree/2022/jul/11/the-guardian-view-on-th

[82] *United Laboratories*, https://www.unitedlabsinc.com/

[83] Bush, Vannevar: *As we may think*, The Atlantic, July 1945, https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/

[84] Wardle, Claire: *A new World Disorder*, Scietific American, 29-4, Fall 2020

[85] Wahlquist, Calla: *US attacks Australia's 'extraordinary' plan to make Google and Facebook pay for news*, The Guardian, 18 Jan 2021, https://www.theguardian.com/media/2021/jan/19/us-attacks-australias-extraordinary-plan-to-make-google-and-facebook-pay-for-news

[86] Wikipedia *Pegasus (spyware)*, https://en.wikipedia.org/wiki/Pegasus_(spyware)

[87] Wikipedia *Timeline of file sharing*, https://en.wikipedia.org/wiki/Timeline _of_file_sharing

[88] Wikipedia *History of email*, https://en.wikipedia.org/wiki/History_of_emai

[89] Wikipedia *Linux for mobile devices*, https://en.wikipedia.org/wiki/Linux _for_mobile_devices

[90] Wikipedia *Simple Mail Transfer Protocol*, https://en.wikipedia.org/wiki/Simple_Mail _Transfer_Protocol

[91] Wikipedia *History of the Internet*, https://en.wikipedia.org/wiki/History _of_the_Internet

[92] Wikipedia *Hypertext Transfer Protocol*, https://en.wikipedia.org/wiki/Hypertext _Transfer_Protocol

[93] Wikipedia *HTML*, https://en.wikipedia.org/wiki/HTML

[94] Wikipedia *Mosaic (web browser)*, https://en.wikipedia.org/wiki/Mosaic _(web_browser)

[95] Wikipedia *List of mergers and acquisitions by Alphabet*. https://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions _by_Alphabet

[96] Wikipedia *List of mergers and acquisitions by Meta Platforms*. https://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions _by_Meta_Platforms

[97] Wikipedia *List of mergers and acquisitions by Microsoft*. https://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions _by_Microsoft

[98] Wikipedia *SoftQuad Software*, https://en.wikipedia.org/wiki/SoftQuad_Software

[99] Wikipedia *List of mergers and acquisitions by Apple*. https://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions_by_Apple

[100] Young, Chris: *Tim Hortons proposes settlement in class-action suits over data-tracking app - Proposed settlement would offer free coffee and doughnut to affected users*, The Canadian Press, Jul 29, 2022, https://www.cbc.ca/news/business/tim-hortons-app-1.6536175

[101] Zakrzewski, Cat; De Vynck, Gerrit; Masih, Niha; Mahtani, Shibani: *How Facebook neglected the rest of the world, fueling hate speech and violence in India*, Washington Post, 24 October, 2021,

## Acknowledgement

The author would like to acknowledge the valuable discussions with members of the family and the contribution of many philosophers, researchers and investigative journalists cited and perhaps missed; these have been valuable in preparing this article.

# Provenance in Spatial Queries

Paulo Pintor
IEETA - University of Aveiro
Aveiro, Portugal
paulopintor@ua.pt

Rogério Luís de C. Costa
CIIC, Polytechnic of Leiria
Leiria, Portugal
rogerio.l.costa@ipleiria.pt

José Moreira
IEETA - University of Aveiro
Aveiro, Portugal
jose.moreira@ua.pt

## ABSTRACT

Despite data growth being a known problem for several years, there are more and more people, tools and devices to create and share data, and the need for tools to infer their provenance and quality is even more important than before. Research on data provenance focuses on W3C PROV and databases (where, why, how). However, in the particular case of spatial data, research has mainly focused on handling spatial data provenance from documents and workflows, but there is no literature approaching the topic of spatial data provenance in DBMS and queries.

This paper deals with the computation of *How–*, *Why–* and *Where–* provenance in spatial database queries. It presents an evaluation of how the formalism and methods proposed to deal with general-purpose database queries behave when dealing with spatial data. Two tools are used to manage provenance in databases and a discussion of the results and guidelines for future work are presented. This is a first contribution towards dealing with spatial data provenance by tuple, attribute and query, whereas previous work has only focused on the management of provenance at a coarser level, namely documents and workflows.

## CCS CONCEPTS

• **Theory of computation → Data provenance**; • **Information systems → Query languages**; **Spatial-temporal systems**.

## KEYWORDS

Data provenance, Spatial data & Queries

**ACM Reference Format:**
Paulo Pintor, Rogério Luís de C. Costa, and José Moreira. 2022. Provenance in Spatial Queries. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3548785.3548802

## 1 INTRODUCTION

Technological evolution is constantly increasing the capacity to obtain and process data from a variety of data sources, such as satellite and aerial images and GPS data captured using mobile devices, among many others. Often, these data have a spatial component that is important to consider. Since these data may originate from many sources, several issues arise regarding the data quality, reliability, and trustworthiness, thus leading us to data provenance and how to explain the origin and transformations made to the data over time.

Besides helping to ensure these features, data provenance may also aid in data debugging by showing how and why a result is obtained, and thus help to catch errors. The information also helps reproducibility or replication, for instance, when we want to perform new tests with a dataset used in the past and we need to recreate the environment [17]. Since it collects data about the transformation and how the result has been obtained, it will also contribute to understandability [17].

The PROV-W3C [8] is a standard for representing and exchanging data about the agents and the processes involved in creating a data instance, using concepts such as Agent, Activity, and Entity [7]. There are also works on representing data provenance in databases with finer granularity, namely, at tuple and attribute level [4, 6, 30] rather than at entity or table level. The main types of provenance in this context are *how-*, *where-* and *why*-provenance, and each one aims to explain the query results from different perspectives.

For spatial data, there are also several works on maintaining information about the origin and transformation of spatial datasets, including the algorithms and computational systems used, from their acquisition to their storage in databases or other types of spatial data infrastructures [20]. However, to the best of our knowledge, there is no work evaluating whether the theories and formalism proposed in the literature to compute the provenance of the results of database queries can also be applied to spatial data.

This paper presents a study on how to compute the provenance of the data in the result of a spatial database query. Starting from a classification of spatial data and operations [26] that allows us to list the types and the cases that we must consider when dealing with spatial queries, we present a systematic evaluation of how to compute and what are *how-*, *where-* and *why*-provenance. This evaluation uses ProvSQL, a tool for provenance and probability management in PostgreSQL [30] and a tool for provenance computation in distributed databases [25].

The rest of this paper is organized as follows. Section 2 presents the basics of data provenance and the formalism used in databases, what spatial data are and what kind of spatial operations exist, and an overview of the spatial data provenance literature. It also highlights that there are research gaps in this area. Next, section 3 shows how to compute the three types of provenance considered in this work for each type of spatial operation and what particularities emerge in this context. Section 4 presents spatial queries and provenance information of the results obtained using the two tools considered in this work.Section 5 presents a brief discussion of the results and guidelines for future work. briefly discusses the results and describes guidelines for future work.

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548802

# 2 BACKGROUND AND RELATED WORK

## 2.1 Data provenance

The concern about data description is not new. In the 1990s, still under the lineage concepts, research has already been made on this subject [5]. Later, in the 2000s, many papers on data provenance in fields like data warehousing, semantic web, and scientific workflows, among others [31], have been published. Recently, more emphasis has been given to this topic due to data science, the problem of data growth, and the need to extract knowledge from raw data and ascertain the data quality.

In [17, 18] it is proposed to divide provenance into four levels, namely, provenance meta-data, information system provenance, workflow provenance, and data provenance. The provenance meta-data is the lowest level in instrumentation and is more general in process or provenance models because it can be anything that describes an object. On the other side, we have data provenance that is more specific in terms of process and model, and with a high level of instrumentation since it collects provenance from databases and depends on the semantics, the query languages, and the data models.

The World Wide Web Consortium (W3C) has proposed a standard model and ontology to describe provenance called PROV to help to describe Workflow provenance [8]. This standard can be seen as an evolution of the Lineage standards ISO 19115 and ISO 19115-2. The W3C PROV contains three different elements: Entity, Activity, and Agent.

- Entity is what we can call a thing, can be real or imaginary, e.g., something digital;
- Activity occurs in a given temporal space. It can use, modify, process or generate new entities;
- An agent can be anything from a person to software. It can have some responsibility for an entity or other agent's existence and some type of responsibility for activities. This responsibility further means that an agent can be a particular type of entity or activity. Hence, we can reveal the provenance of an agent.

Although agreeing with the W3C proposed model, some authors mention that in a few database contexts, the model brings some unnecessary formalisms [5]. In contrast, it is also possible to understand the model's potential when applied in complex areas such as health and geospatial data [8, 21].

The main types of data provenance in the literature are *why-*, *how-* and *where-provenance* [5, 6, 14].

*Why-provenance* gives the tuples that contributed to a query result [4–6, 14]. The tuples are seen as witnesses of the result, and the technique to obtain this kind of provenance is called Witnesses basis. Formally, it can be described as $Why(Q, I, t) = \{I' \subseteq I | t \in Q(I')\}$ such that, for a query $Q$ over a database $I$ and a tuple $t$ in *Q(I)*, an instance of $I' \subseteq I$ is a witness for $t$ if $t \in Q(I')$ [4, 6]. The result is a set of tuples with all the possible combinations but does not contain duplicates.

Whilst *Why-provenance* shows the tuples that are involved in a query result, *How-provenance* explains how these tuples contributed to the result. *How-provenance* resorts to algebraic and polynomials expressions, called semirings, to obtain the result, and this technique assumes that each tuple has an identifier called prove token [4–6, 13, 14, 28].

The general definition of a semiring is: $(K, 0, 1, \oplus, \otimes)$, and can be used to achieve different answers. For *how-provenance* the universal semiring or how-semiring is $(N[X], 0, 1, \oplus, \otimes)$. The $N$ is a set of data elements that will be annotated using the constants 0 and 1, where 0 means the tuple is not in the query result $Q$ and 1 means the tuple is present in the query result. The binary operators show how tuples relate to each other, such that $\oplus$ is used to show an alternative and $\otimes$ a joint. The semiring has commutative, associative, and distributed rules, depending on the query's operation [13]. *Unions* are associative and commutative operations and are represented by $\oplus$. *Joins* also have those two properties, but they are also distributive over *unions* and they are represented by $\otimes$. The *projections* and *selections* are also commutative among themselves.

From the above, *why-provenance* gives the tuples (witnesses) involved in a result and *how-provenance* explains how the tuples correlate to get the result. The last one, *where-provenance*, is concerned with the origin of the individual values (instead of tuples) in the query result. Thus, Unlike the previous ones, it is not possible to capture this kind of provenance using semirings. To deal with this issue, [28] proposes to add annotations without algebra terms to create a bipartite graph that shows how the values are connected to specify the *where-provenance* result.

## 2.2 Spatial data

In current systems, data can come from a variety of sources, which can be more or less reliable, and data cleaning and transformation processes, including data integration and wrangling, also have an influence on data accuracy and quality [1]. These issues above are transverse to all data types. However, they are particularly important when dealing with spatial data.

Spatial data are used to represent geographical objects, such as roads, properties, and lakes.In the context of geographic information systems, spatial data are represented using coordinates (latitude and longitude). The three main types of spatial data are point, line, and polygon (often also referred as region) [15, 16, 27],

- A point represents a specific location, like a traffic light in a street;
- A line can be curved and represent roads, for example;
- A region is delimited by external boundaries and can have or not internal boundaries to represent holes. It can define objects with an extension like a lake.

Moreover, spatial objects can also be represented by more complex geometries, e.g., multi-points or multi-polygons, and even geometries combining points, lines, and polygons. There is a standard data model called SQL/MM Spatial [32] to represent spatial data in Database Management Systems (DBMS). This standard was known initially as the OpenGIS Simple Features Specification for SQL and defines the spatial data formats, including the well-known text representation (WKT), well-known binary representation (WKB), and geography markup language (GML), as well as the Spatial Reference Systems. SQL/MM also defines the operations and functions to deal with spatial data, namely, to convert between spatial data formats, retrieve spatial properties, and find the interaction between spatial objects.

In this work, we will use the classification of spatial operations proposed in [26]:

- Regarding the number of parameters, the spatial operations can be unary or N-Ary. The former accepts only one parameter, e.g., to return the area of a polygon or the length of a line, to transform an object (rotation, translation, skewing, and scaling) or to simplify an object. Binary operations accept two parameters and can be further classified into topological (e.g., *adjacent*, *contains*, *inside*, *equals* or *overlaps*), distance or direction (e.g., *North*, *South*, *Northwest*, etc.). These operations are often called spatial joins. Set operations can operate on multiple spatial objects, e.g., the union of several polygons.
- Spatial operations can return a Boolean, a value, or new spatial objects. The former are called predicates and can be unary, e.g., *isValid* to determine if the geometry of a spatial object is valid, or binary to check if there is a topological or directional relationship between two spatial objects. Other operations return a scalar, such as an area, perimeter, or distance operations. Finally, there are operations that return new spatial objects and can be unary, e.g., to return the convex-hull of a spatial object, binary, e.g., to obtain the difference between two spatial objects, or operate on several spatial objects, e.g., to compute the union or intersection of spatial objects.

## 2.3 Related Work

Spatial data may have distributed sources and the data sets can be unstructured (e.g. a file). Therefore, it is important to ensure the datasets' quality and trustworthiness, bearing in mind problems such as the completeness of the data sets or data set decadence over the years [23]. The last problem is widespread in this data type since it can represent maps. For example, roads are constantly changing in a city (road works, directions change, etc.), so a dataset that dates back a few years may not correspond to a road's actual state.

Consequently, provenance is essential to ensure the quality of the data and its trustworthiness. Looking at the literature on spatial data provenance, it is possible to understand that the focus is on Workflows. Most recent works use the W3C PROV [7, 8], while early works in this field started with the Lineage standards of ISO 19115 and ISO 19115-2 [9, 10]. A general transition is being made from ISO standards to W3C PROV, as demonstrated in [19, 20] where it is possible to understand how to proceed with the transition.

Spatial workflows are used to store and show the information about all the steps or processes applied to the data. This includes the data's origin and the operations or algorithms that the data have been through until the final dataset. An excellent example of this process can be a map composed of multiple layers. Resorting to the W3C elements - agents, entities, and activities - is possible to store the multiple data's origins, demonstrate if some algorithm made some change to the data (like coordinators correction), and the procedures to create the layers formed by attributes or characteristics, and the final dataset representing the map.

There also are works trying to use provenance with unstructured spatial data from text [22]. In this case, the authors used data from social networks (location attributes, to be more precise) and studied the possibility of deriving the provenance of that unstructured information. The authors claim to have 80% accuracy in identifying the location provider.

None of the previous works deal with the provenance of the results in spatial database queries. Although, it is important to understand if the spatial data characteristics and specifications can be supported by the existing approaches in different scenarios.

In [24], the authors conducted a survey over several solutions for provenance in different areas, including data provenance.Though, these solutions to deal with spatial data need at least to work with DBMS that has support for that kind of data. Therefore, combining the research presented in that paper with the objectives of our work, we chose three solutions to show how different the approaches can be. The three solutions are ProvSQL [30], Perm [12] and GProM [3]. These three solutions work with PostgreSQL, a DBMS that supports spatial data, but they all approach the data provenance problem with different perspectives.

Perm is an extension to PostgreSQL, and the approach is based on query rewriting. The authors mentioned that it supports regular queries, although it is not prepared to support correlated subqueries. Perm engages the why- and where-provenance.

GProM is the only one in these three solutions that works with more than one DBMS. In this case, it works with Oracle, SQLite, and PostgreSQL, and the approach is to use a middleware. This platform intends to be used not only for provenance management but also for annotations management. GProM captures the why- and why-not provenance in terms of data provenance, and the authors claimed it can also deal with spatio-temporal information.

ProvSQL is a lightweight extension for PostgreSQL that supports provenance computation and probabilistic query evaluation. ProvSQL uses semiring theory to compute how-provenance and proposes an extension to semirings called m-semirings to support negation. It also supports the capture of where-provenance. This work, like the two mentioned above, does not show how to deal with data provenance and spatial queries.

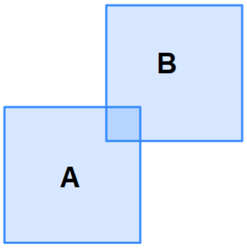
# 3 SPATIAL PROVENANCE

In this section, we show how SQL/MM Spatial objects and functions are related with data provenance, focusing on operations such as intersection, distance or overlaps, which are not considered in previous work.

In data provenance, it is necessary to understand how these operations act in the query result, and with the example of semiring theory, how the operators $\oplus$ and $\otimes$ should be used with these functions. The Spatial functions can be divided into several different groups according to what operations we need to perform with the data. In the following, we will divide the spatial operations based on [26] and for each type of spatial operations, we investigate how *Why–*, *How–* and *Where–* data provenance can be used.

To also help demonstrate data provenance with spatial objects, we will use the objects depicted in Figure 1 and Table 1. It is assumed that each spatial object has an attribute "geom" denoting its representation (geometry), an attribute "token" that is the token needed to compute the data provenance, and a column "name".

**Figure 1: An example of spatial objects**



**A**

| name | geom | token |
|---|---|---|
| A | POLYGON((2.5 2.5,2.5 4.5,4.5 4.5,4.5 2.5,2.5 2.5)) | t1 |
| B | POLYGON((4 4,4 6,6 6,6 4,4 4)) | t2 |

**Table 1: Table representing the two objects from figure 1 with the name "A"**

| name | how | why | where |
|---|---|---|---|
| A | (t1) | {t1} | {[A.t1.name]} |
| B | (t2) | {t2} | {[A.t2.name]} |

**Table 2: Result provenance for Unary Operations with boolean result.**

| name | area | how | why | where |
|---|---|---|---|---|
| A | 4 | (t1) | {t1} | {[A.t1.name], []} |
| B | 4 | (t2) | {t2} | {[A.t2.name], []} |

**Table 3: Result provenance for Unary Operations with scalar result.**

## 3.1 Unary operations

Unary operations are spatial operations involving only one object and can be divided according to the result.

*Operations with boolean result.* - These functions will return a boolean result for a test in a particular object. Looking at Figure 1 and Table 1, an example of a query could be to select a column with the objects' name ("name") and a column indicating whether the geometry of each of the selected objects is valid, resorting to the function "isvalid" and the polygons geometry ("geom") column. The three types of provenance of the query's result are presented in Table 2.

The result shows that for *why*- and *how*-provenance, the provenance information is formed only by the token of the tuples, and the *where*-provenance also includes the table identifier, the tuple token and the respective column.

*Operations with scalar result.* - functions with scalar results aim to obtain values like area, perimeter or length of an object. To show the provenance results, we will calculate the area of the two squares and show it with the name.

| name | rotate | how | why | where |
|---|---|---|---|---|
| A | POLYGON((-2.5 -2.5,-2.5 -4.5,-4.5 -4.5, -4.5 -2.5,-2.5 -2.5)) | (t1) | {t1} | {[A.t1.name], []} |
| B | POLYGON((-4 -4,-4 -6,-6 -6,-6 -4,-4 -4)) | (t2) | {t2} | {[A.t2.name], []} |

**Table 4: Result provenance for Unary Operations with spatial result.**

Table 3 shows that for the functions with scalar results, *how*- and *why*-provenance are the tuple tokens. But for *where-provenance*, it shows the set of tuples with the tables for each column in the output result, although the "area" column has the result of ∅. This is because *where*-provenance is attributed-based, in contrast to the other types, which are tuple-based. The new column, created by the function, does not exist in the database and has no meaning for *where*-provenance.

*Operations with spatial result.* - functions with a spatial result intended to transform one spatial object into another, and there are many different operators. One example with the squares in the Figure 1 can be the rotation of each polygon by Π radius - rotate(geom, pi()).

As it is possible to understand by the result in Table 4, the *how*- and *why*-provenance are the tuples tokens, and *where*-provenance only shows provenance for the name since the rotations create a new object that is not part of the table.

## 3.2 Binary operations

These operations always involve two objects, and as the unary operations, they can be divided by the same result types.

*Operations with boolean result.* - In binary, these types of operations are called Spatial Predicates. They are the basis for spatial querying (Spatial selections and Spatial joins) [26]. These operations have three subdivisions: Topological predicates, direction predicates and metric predicates. The first one represents topological transformations like intersects or contains, among others and shows, for example, if a line intersects a point. The second shows the relative position between two objects, that is, whether one object is south of another. The metric predicates can be used to compare the distance between objects.

Consider Figure 1 and Table 1. To intersect the two rectangles, one may perform a query that joins the table with itself by joining polygons with different names that intersect each other.

| name | name | how | why | where |
|---|---|---|---|---|
| A | B | (t1 ⊗ t2) | {t1, t2} | {[A.t1.name, A.t2.name], [A.t1.name, A.t2.name]} |
| B | A | (t2 ⊗ t1) | {t2, t1} | {[A.t1.name, A.t2.name], [A.t1.name, A.t2.name]} |

**Table 5: Result provenance for Binary Operations with boolean result.**

Table 5 shows the results of binary operations with boolean results and demonstrates that the results are different when compared to the unary operations with boolean results (Table 2) in terms of *how-provenance*. Since it involves two objects, a binary operation needs an operator, and the intersect can only be formed with both. Thus the ( ⊗ ) is used . *Why-provenance* has one set of tuples that correspond to both tuples' tokens, and *Where-provenance* shows two tokens with the table for each column. Since it is a self join, the where-provenance is equal for both columns involved in the join and both tuples.

*Operations with scalar result.* - In unary operations, a scalar result involves measures for one object. In binary operations, these operations will also involve measures between two objects, like the function distance.

To calculate the distance, we need to join the table with itself, as in the previous example, although, as we see in the result, the distance between the objects is zero because they intersect each other.

| name | name | distance | how | why | where |
|---|---|---|---|---|---|
| A | B | 0 | (t1 ⊗ t2) | {t1, t2} | {[A.t1.name, A.t2.name],<br>[A.t2.name, A.t1.name]} |
| B | A | 0 | (t2 ⊗ t1) | {t2, t1} | {[A.t1.name, A.t2.name],<br>[A.t1.name, A.t2.name]} |

**Table 6: Result provenance for Binary Operations with scalar result.**

The provenance results in Table 6 are similar to the previous, with the difference we already saw in unary operations. The "distance" column has no *where*-provenance.

*Operations with spatial result.* - these operations use two different objects to create a new one. If one thinks of examples with roads and maps, these operations can be seen as the basis for map layers because they allow the creation of layers over layers with functions such as union or difference [26]. Examples of these functions are intersection and union, among others.

The intersection will be the query to show the data provenance in this last operation. In this query we will join the table with itself, and we will filter by the square with the name "A" in the first table and the square with the name "B" in the second. The query result will be the small square where the two squares overlap.

The *how*- and *why*-provenance are again the two tokens, and since we only selected the intersection, the *where*-provenance is ∅.

Therefore, the unary operations have as provenance result the tuple token since they only involve one object. The binary operators are different since they can be spatial joins or need a join to allow the operations. Thus, they need an operator in *how*-provenance. *Where*- and *why*-provenance has more then one tuple in each set.

| polygon | how | why | where |
|---|---|---|---|
| POLYGON ((4.5 4.5, 4.5 4, 4 4, 4 4.5, 4.5 4.5)) | (t1 ⊗ t2) | {t1, t2} | {[]} |

**Table 7: Result provenance for Binary Operations with spatial result.**

It is essential to refer to the where-provenance, all the functions used in the projection and create a new column the where-provenance is ∅ as mentioned above.

Spatial functions also have other functions to help create, alter objects or add columns to the tables. Although, those functions are like the updates or alters in standard. These functions are not involved in selections, hence they are not part of data provenance, but there are studies to understand the schema and data changes and how to use provenance to help keep track of these changes [11].

# 4 EXPERIMENTAL EVALUATION

This section will explain our test environment, where we will explain the dataset used and the solutions used to perform the tests. We will also explain the tests performed and the results obtained, and we will finish the section with a discussion about the results obtained.

## 4.1 Environment

The database chosen for our test environment is PostgreSQL for two main reasons. The first reason is because it is easy to deal with spatial objects using the PostGIS extension. The second reason is that one of the solutions we will present to generate the data provenance information was implemented on PostgreSQL.

The dataset used is demonstrated in tables 8 and 9. The table 8 with the name "B" has only polygons, most specifically one triangle and three squares, as it is possible to see in figures 2. Whereas the table 9 with the name "C" has only a polygon (a square), two points and a line as shown in figure 3. Both tables have three columns, the column "name" to help to identify objects in the queries, the "geom" column, which is the objects' geometrical representation and the token needed for the data provenance.

**B**

| name | geom | token |
|---|---|---|
| A | POLYGON((0 0,0 6,6 0,0 0)) | t1 |
| B | POLYGON((4 4,4 6,6 6,6 4,4 4)) | t2 |
| C | POLYGON((4 0,4 2,6 2,6 0,4 0)) | t3 |
| D | POLYGON((6 5,6 7,8 7,8 5,6 5)) | t4 |

**Table 8: Table B**

**C**

| name | geom | token |
|---|---|---|
| E | POLYGON((0 0,0 2,2 2,2 0,0 0)) | t5 |
| F | POINT(5.5 4.5) | t6 |
| G | LINESTRING(3 7,5 5) | t7 |
| H | POINT(6 1) | t8 |

**Table 9: Table C**

We experimentally evaluated the use of two data provenance solutions in order to understand if the solutions and theories behind the provenance types work with spatial functions.

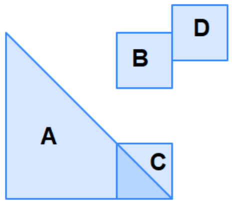


**Figure 2: Objects from Table B**

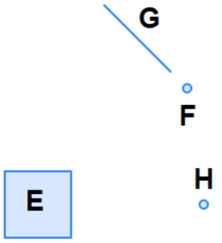


**Figure 3: Objects from Table C**

| | Soltuions | |
|---|---|---|
| Data provenance types | ProvSQL | Distributed |
| Why | ✗ | ✓ |
| How | ✓ | ✓ |
| Where | ✓ | ✗ |
| Probabilistic | ✓ | ✗ |

**Table 10: Resume of solutions and the provenance types they accept**

The first solution is ProvSQL [30] a lightweight extension for PostgreSQL. This application not only deals with *how*- and *where*-provenance but also allows probabilistic queries. It also shows what the author call m-semiring to deal with non-monotone queries.

The second solution ([25]) has a different approach to dealing with data provenance. First, it works independently of the databases, and second, it can be used in distributed environments. This approach has two different models. One module will proceed with the query re-writer, and the second module will build the provenance information based on the annotations with provenance tuples added by the query re-writer. This second solution will also be applied over the same PostgreSQL database used in the ProvSQL solution. This second approach will give us the *why*-provenance allowing us to incorporate all the three main data provenance types in our tests.

Regarding querys' syntax, the solutions have different syntaxes to allow the provenance information in the final query result. In the ProvSQL solution, since it is an extension to PostgreSQL, the user needs to write the new data provenance's function syntax in the query depending on what type of data provenance wants to obtain, and also needs some pre-processing. The authors provide a helpful and straightforward guide to creating the tokens, mapping the tokens, and using the data provenance functions.

The second solution re-writes the query by adding the data provenance syntax without the user's concern. In contrast with ProvSQL, this solution does not add functions to the query syntax. Instead, it adds columns with the tokens divided by different delimiters, allowing the second model to build the information. This solution assumes that the tables already had the tokens.

In terms of data provenance syntax in the final result, both solutions use the same syntax as also used in the definitions of the data provenance types [4, 6]

| | Operations | |
|---|---|---|
| Result | Unary | Binary |
| boolean | ST_IsValid(geom) | ST_Intersects(geom,geom) |
| scalar | ST_Area(geom) | ST_Distance(geom,geom) |
| spatial | ST_Rotate(geom,float) | ST_Intersection(geom,geom) |

**Table 11: Resume the functions used in the tests by operation and result**

| Distance | how | why | where |
|---|---|---|---|
| 11.180339887498949 | (t2 ⊗ t8) | {t2, t8} | [{}] |
| 13.416407864998739 | (t4 ⊗ t8) | {t4, t8} | [{}] |

**Table 12: Results of Query 1**

As it is possible to understand with the table 10, to have the three main data provenance types, we need to use both solutions. The distributed solution is a new approach and, for now, cannot deal with the where-provenance and probabilistic, but as mentioned by the authors, it is something for future work. The where-provenance is not tuple-based and needs different approaches to be built. The probabilistic queries can be a challenge because looking at the ones used in [30] (c2d, d4, dsharp, weightmc, graph-easy), it is necessary to understand how to apply them in not only a distributed environment and if all the involved structures support it.

## 4.2 Experimental work and results

The tests performed over our datasets were thought to cover the different types of binary and unary operations. Table 11 shows a summary of what operations will be used for each type.

The first experience will be with Query 1. It will include a unary operation with a spatial result and a binary operation with a scalar result. The query will rotate the squares "B" and "D" from Table 8 and calculate the distance between the rotated squares to the point "H" in Table 9.

**Listing 1: A query with rotation and distance**

```
SELECT ST_Distance(NewB.geom, C.geom) as Distance
FROM
(
    SELECT ST_Rotate(geom, PI()) as geom
    FROM B
    WHERE name = 'B' or name = 'D'
) NewB, C
WHERE C.name = 'H'
```

Table 12 shows that *How*-provenance has two tuples and the ⊗ operator, meaning that we need both tuples to obtain a row. *Why*-provenance has one set of two witnesses, and *where*-provenance is empty since the distance is a calculated column.

The next test will be with a query that will use a unary operator with a boolean result and a binary operator with the same type of result. Query 2 joins the two tables resorting to the binary operator intersects, and the output will show if the two polygons that intersect each other are valid.

| name | av | name | bv | how | why | where |
|---|---|---|---|---|---|---|
| A | true | E | true | (t1 ⊗ t5) | {t1, t5} | {[B:t1:1], [],<br>[C:t5:1], []} |
| B | true | F | true | (t2 ⊗ t6) | {t2, t6} | {[B:t2:1], [],<br>[C:t6:1], []} |
| B | true | G | true | (t2 ⊗ t7) | {t2, t7} | {[B:t2:1], [],<br>[C:t7:1], []} |
| C | true | H | true | (t3 ⊗ t8) | {t3, t8} | {[B:t3:1], [],<br>[C:t8:1], []} |

**Table 13: Results of Query 2**

| Area | how | why | where |
|---|---|---|---|
| 4 | (t1 ⊗ t5) | {t1, t5} | {[]} |

**Table 14: Result of the Query 3**

**Listing 2: A query with intersects and IsValid**

```
SELECT B.name, ST_IsValid(B.geom) as BV,
C.name, ST_IsValid(C.geom) as CV
FROM B, C
WHERE ST_intersects(B.geom, C.geom)
```

Table 13 shows that the two solutions can deal with spatial joins and present the provenance result. In terms of *how*- and *why*-provenance, the result is similar to the previous one. However, we can see a difference in *where*-provenance. *Where*-provenance has results for two columns, in this case for the "name" columns, the other two are ∅. In ProvSQL, the authors represent *where*-provenance with the table in the first place, then the token and in the end, the number is the column index in the table. The columns "name" are the first column in both tables.

The next experience is represented by the Query 3, and it uses a unary scalar result operation and a binary operation with a spatial result. It calculates the area of the new object resulting from the show the intersection between polygon "A" in table 8 and the polygon "E" in table 9.

**Listing 3: A query with the area of a new object**

```
SELECT ST_Area(ST_Intersection(B.geom, C.geom))
FROM B, C
WHERE B.name = 'A'
  AND C.name = 'E'
```

In Section 3.1, we demonstrated that when we use the area function alone, the provenance result has just one tuple, i.e., the row tuple. Although, in Table 14, we can see that the *how*- and *why*-provenance have two tuples, and the first one is conjugate by the ⊗. The binary ("Intersection") and the unary ("Area") functions, if used independently from each other, would give different results. The *how*- and *why*-provenance for the "Area" would be only the tuple token, and for the binary, it would be the conjugation of two tokens because we were joining (intersecting) two polygons. The result in 14 demonstrates that when we applied both simultaneously, the joining data provenance prevails over the unary. It is also possible to understand that since the column presented is created by the functions, the *where*-provenance is not applied.

## 4.3 Results discussion

The results of our tests demonstrated that the solutions work with spatial objects and spatial functions independently of type (unary or binary). Consequently, it demonstrates that data provenance theories can be applied to different data types from the standard.

Although, some spatial functions can have variations that use more than two values, and the input can also receive an array or an aggregation. In the aggregation, these functions will behave as standard functions, such as the average of values (AVG). One of these operators is the union. Unfortunately, none of the solutions supports, for now, this type of aggregation.

As stated in [29], the aggregation operators need a paradigm change from semirings to semimodules. In [2], is an example of a study on how to use the semimodules to represent that kind of operator. However, there is still no practical work with a solution applying the semimodules.

# 5 CONCLUSION

In this work, we addressed the topics of data provenance and spatial data.

We presented the basic concepts about data provenance and spatial databases and a brief literature review showing that combining these two subjects is still an open research topic.

Next, we presented a study on how to compute and interpret data provenance results based on a classification of spatial operations and three main types of data provenance (*how*-, *why*- and *where*-provenance). The results show that these tools, which were developed for managing provenance in general-purpose (non-spatial) databases, are also capable of handling different types of spatial operations, namely predicates, topological operations, and spatial joins, among others. The exceptions are aggregations and set operations, for instance, to compute the union of spatial objects.

In future work, we would perform more tests with data provenance but now with Spatio-temporal data to understand how temporal queries can affect the data provenance and if we can collect the data provenance from such specific operations as the temporal operations.

# ACKNOWLEDGMENTS

Paulo Pintor has a research grant awarded by the Portuguese public agency for science, technology and innovation FCT - Foundation for Science and Technology - under the reference 2021.06773.BD. This work is partially funded by National Funds through the FCT under the Scientific Employment Stimulus - Institutional Call - CEECIN-ST/00051/2018, and in the context of the projects UIDB/04524/2020 and UIDB/00127/2020.

# REFERENCES

[1] Daniel Abadi, Anastasia Ailamaki, David Andersen, Peter Bailis, Magdalena Balazinska, Philip Bernstein, Peter Boncz, Surajit Chaudhuri, Alvin Cheung, AnHai Doan, Luna Dong, Michael J. Franklin, Juliana Freire, Alon Halevy, Joseph M. Hellerstein, Stratos Idreos, Donald Kossmann, Tim Kraska, Sailesh Krishnamurthy, Volker Markl, Sergey Melnik, Tova Milo, C. Mohan, Thomas Neumann, Beng Chin Ooi, Fatma Ozcan, Jignesh Patel, Andrew Pavlo, Raluca Popa, Raghu Ramakrishnan, Christopher Ré, Michael Stonebraker, and Dan Suciu. 2020. The Seattle Report on Database Research. *SIGMOD Rec.* 48, 4 (feb 2020), 44–53. https://doi.org/10.1145/3385658.3385668

[2] Yael Amsterdamer, Daniel Deutch, and Val Tannen. 2011. Provenance for Aggregate Queries. In *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART*

*Symposium on Principles of Database Systems* (Athens, Greece) *(PODS '11)*. Association for Computing Machinery, New York, NY, USA, 153–164. https://doi.org/10.1145/1989284.1989302

[3] Bahareh Sadat Arab, Su Feng, Boris Glavic, Seokki Lee, Xing Niu, and Qitian Zeng. 2018. GProM - A Swiss Army Knife for Your Provenance Needs. *IEEE Data Engineering Bulletin* 41, 1 (2018), 51–62. http://sites.computer.org/debull/A18mar/p51.pdf

[4] Peter Buneman, Sanjeev Khanna, Wang-Chiew Tan, and Wang Chiew. 2001. Why and Where: A Characterization of Data Provenance. *Computer Science* 1973 (2001), 316–330.

[5] Peter Buneman and Wang Chiew Tan. 2018. Data provenance: What next? *SIGMOD Record* 47, 3 (2018), 5–16. https://doi.org/10.1145/3316416.3316418

[6] James Cheney, Laura Chiticariu, and Wang Chiew Tan. 2007. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases* 1 (2007), 379–474. Issue 4. https://doi.org/10.1561/1900000006

[7] Guillem Closa, Joan Masó, Núria Julià, and Xavier Pons. 2021. Geospatial Queries on Data Collection Using a Common Provenance Model. *ISPRS International Journal of Geo-Information* 10 (2021), 139. Issue 3. https://doi.org/10.3390/ijgi10030139

[8] Guillem Closa, Joan Masó, Benjamin Proß, and Xavier Pons. 2017. W3C PROV to describe provenance at the dataset, feature and attribute levels in a distributed environment. *Computers, Environment and Urban Systems* 64 (2017), 103–117. https://doi.org/10.1016/j.compenvurbsys.2017.01.008

[9] Guillem Closa, Joan Masó, Alaitz Zabala, Lluís Pesquer, and Xavier Pons. 2019. A provenance metadata model integrating ISO geospatial lineage and the OGC WPS: Conceptual model and implementation. *Transactions in GIS* 23 (2019), 1102–1124. Issue 5. https://doi.org/10.1111/tgis.12555

[10] Liping Di, Yuanzheng Shao, and Lingjun Kang. 2013. Implementation of geospatial data provenance in a web service workflow environment with ISO 19115 and ISO 19115-2 lineage model. *IEEE Transactions on Geoscience and Remote Sensing* 51 (2013), 5082–5089. Issue 11. https://doi.org/10.1109/TGRS.2013.2248740

[11] Shi Gao and Carlo Zaniolo. 2012. Supporting Database Provenance under Schema Evolution. In *Advances in Conceptual Modeling* (Florence, Italy) *(ER'12)*. Springer-Verlag, Berlin, Heidelberg, 67–77. https://doi.org/10.1007/978-3-642-33999-8_9

[12] Boris Glavic and Gustavo Alonso. 2009. Perm: Processing provenance and data on the same data model through query rewriting. In *Proceedings of the International Conference on Data Engineering*. IEEE, Shanghai, China, 174–185. https://doi.org/10.1109/ICDE.2009.15

[13] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. 2007. Provenance Semirings. In *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Beijing, China) *(PODS '07)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/1265530.1265535

[14] Todd J. Green and Val Tannen. 2017. The Semiring Framework for Database Provenance. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* (Chicago, Illinois, USA) *(PODS '17)*. Association for Computing Machinery, New York, NY, USA, 93–99. https://doi.org/10.1145/3034786.3056125

[15] Ralf Hartmut Güting. 1994. An Introduction to Spatial Database Systems. *The VLDB Journal* 3, 4 (oct 1994), 357–399.

[16] Ralf Güting, Michael Böhlen, Martin Erwig, Christian Jensen, Nikos Lorentzos, Enrico Nardelli, Markus Schneider, and José Viqueira. 2003. Spatio-temporal models and languages: An approach based on data types. *Lecture Notes in Computer Science* 2520 (01 2003), 117–176.

[17] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. 2017. A survey on provenance: What for? What form? What from? *VLDB Journal* 26, 6 (2017), 881–906. https://doi.org/10.1007/s00778-017-0486-1

[18] Melanie Herschel and Marcel Hlawatsch. 2016. Provenance: On and behind the screens. *Proceedings of the ACM SIGMOD International Conference on Management of Data* 26-June-2016 (2016), 2213–2218. https://doi.org/10.1145/2882903.2912568

[19] I. Ivánová, K. Armstrong, and D. McMeekin. 2017. Provenance in the next-generation spatial knowledge infrastructure. *Proceedings - 22nd International Congress on Modelling and Simulation, MODSIM 2017* (2017), 410–416. Issue December. https://doi.org/10.36334/modsim.2017.c2.ivanova

[20] Liangcun Jiang, Peng Yue, Werner Kuhn, Chenxiao Zhang, Changhui Yu, and Xia Guo. 2018. Advancing interoperability of geospatial data provenance on the web: Gap analysis and strategies. *Computers and Geosciences* 117 (2018), 21–31. Issue May. https://doi.org/10.1016/j.cageo.2018.05.001

[21] Ann Kristin Kock-Schoppenhauer, Lina Hartung, Hannes Ulrich, Petra Duhm-Harbeck, and Josef Ingenerf. 2018. Practical Extension of Provenance to Healthcare Data Based on the W3C PROV Standard. *Studies in Health Technology and Informatics* 253 (2018), 28–32. Issue January. https://doi.org/10.3233/978-1-61499-896-9-28

[22] Kisung Lee, Raghu Ganti, Mudhakar Srivatsa, and Prasant Mohapatra. 2013. Spatio-temporal provenance: Identifying location information from unstructured text. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 499–504. https://doi.org/10.1109/PerComW.2013.6529548

[23] Joana E.G. Malaverri, Claudia Bauzer Medeiros, and Rubens Camargo Lamparelli. 2012. A provenance approach to assess the quality of geospatial data. *Proceedings of the ACM Symposium on Applied Computing* (2012), 2043–2044. https://doi.org/10.1145/2245276.2232116

[24] Beatriz Pérez, Julio Rubio, and Carlos Sáenz-Adán. 2018. A Systematic Review of Provenance Systems. *Knowl. Inf. Syst.* 57, 3 (dec 2018), 495–543. https://doi.org/10.1007/s10115-018-1164-3

[25] Paulo Pintor, Rogério Luís de Carvalho Costa, and José Moreira. 2022. Why- and How-Provenance in Distributed Environments. In *Database and Expert Systems Applications*, Christine Strauss, Alfredo Cuzzocrea, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil (Eds.). Springer International Publishing, Cham, 103–115.

[26] Philippe Rigaux, Michel Scholl, and Agnès Voisard. 2001. Spatial Databases with Application to GIS. *SIGMOD Record* (01 2001).

[27] Markus Schneider. 2009. *Spatial and Spatio-Temporal Data Models and Languages*. Springer US, Boston, MA, 2681–2685. https://doi.org/10.1007/978-0-387-39940-9_360

[28] Pierre Senellart. 2017. Provenance and probabilities in relational databases: From theory to practice. *SIGMOD Record* 46 (2017), 5–15. Issue 4. https://doi.org/10.1145/3186549.3186551 7, 5.

[29] Pierre Senellart. 2019. *Provenance in Databases: Principles and Applications*. Springer-Verlag, Berlin, Heidelberg, 104–109. https://doi.org/10.1007/978-3-030-31423-1_3

[30] Pierre Senellart, Louis Jachiet, Silviu Maniu, and Yann Ramusat. 2018. ProvSQL: Provenance and probability management in PostgreSQL. *Proceedings of the VLDB Endowment* 11, 12 (2018), 2034–2037. https://doi.org/10.14778/3229863.3236253

[31] Umber Sheikh, Abid Khan, Bilal Ahmed, Abdul Waheed, and Abdul Hameed. 2018. Provenance Inference Techniques: Taxonomy, comparative analysis and design challenges. *Journal of Network and Computer Applications* 110, March (2018), 11–26. https://doi.org/10.1016/j.jnca.2018.03.004

[32] Knut Stolze. 2003. SQL/MM Spatial - The Standard to Manage Spatial Data in a Relational Database System.. In *BTW*, Vol. 26. 247–264.

# Feature Analysis of Indus Valley and Dravidian Language Scripts with Similarity Matrices

Sarat Sasank Barla
School of Computing, University of Nebraska-Lincoln
sbarla2@huskers.unl.edu

Sai Surya Sanjay Alamuru
School of Computing, University of Nebraska-Lincoln
salamuru2@huskers.unl.edu

Peter Z. Revesz[*]
School of Computing, University of Nebraska-Lincoln
revesz@cse.unl.edu

## ABSTRACT

This paper investigates the similarity between the Indus Valley script and the Kannada, Malayalam, Tamil, and Telugu scripts that are used to write Dravidian languages. The closeness of these scripts is determined by applying a feature analysis of each sign of these scripts and creating similarity matrices that describe the similarity of any pair of signs from two different scripts. The feature list that we use for the analysis of these Dravidian language-related scripts includes six new features beyond the thirteen features that were used for the study of Minoan Linear A and related scripts by Revesz. These new features are the check mark, short vertical line, dot, upper curve, parallel curves, and horizontal line features.

## CCS CONCEPTS

• **Information systems** → Information systems applications; Data mining.

## KEYWORDS

Dravidian, Epigraphy, Feature analysis, Indus Valley script, Script similarity measure, Sumerian pictogram

**ACM Reference Format:**
Sarat Sasank Barla, Sai Surya Sanjay Alamuru, and Peter Z. Revesz. 2022. Feature Analysis of Indus Valley and Dravidian Language Scripts with Similarity Matrices. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3548785.3548801

## 1 INTRODUCTION

It is strongly believed by most of the people that the first human civilization flourished somewhere near the present day upper eastern part of Africa and that all humanity at that time used to speak a single language called a protolanguage, which is the origin of all the languages spoken in today's world [13]. The protolanguage spread and diversified together with human populations as humans started to leave the Sahara when the temperatures started soaring and the desertification of the Sahara begun. The desertification prompted people to split into small groups and to travel to different places in search of food, shelter, and viable climatic conditions. This process resulted in a change in the living style of people along with their environmental needs, requirements, and way of communicating. Although many scientists and researchers believe in the concept of divergence of languages from a protolanguage, this hypothesis is still controversial. Finding how similar two languages is a complex problem. The following are three major ways which help us determine how closely languages are related.

### 1.1 Human migrations

In this method, we try tracking people's migration throughout history and observe how does this migration affected the languages. Generally, the scientists relate linguistics to molecular biology. From the concept of tracking the mitochondria present inside the nucleus of the human body one can trace back people's ancestors, and research suggests this process also works well for finding the language path. However, we cannot completely rely on our process in this method, since when starting to go far back in time we will have less evidence and no accurate metrics on which to base our assumptions.

### 1.2 Similar sounding words

We know that there are many languages that are derived from others which contain the same words which convey similar meanings. However, there is a very high probability of a word with the same sound having a different meaning. These are known as homophones. For example, the word *filter* in English coveys a meaning of a substance which is used to separate different things, but the same word means 'poison' in French. Such words are false cognates. Hence simply looking for similar sounding words is a faulty method.

### 1.3 Feature analysis

In this approach, we find the similarity between two languages by observing the similarity between the scripts and their regular changes. This process is done by developing features which represent all the letters in the scripts and developing the feature evaluation table. When we have the feature analysis tables for at least two languages we can create the similarity matrix to check how close the two scripts are related. We follow this method in our implementation process.

## 2 BACKGROUND

The Indus Valley Script is an ancient script developed by the Indus Valley civilization, which existed c. 3500-1900 BCE. The Indus Valley Civilization was first identified at Harappa and Mohenjo-Daro in 1921 and 1922, respectively [7]. The first publication of the seal with Harappan symbols were produced in 1875 in the drawings of

[*] Corresponding author.

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548801

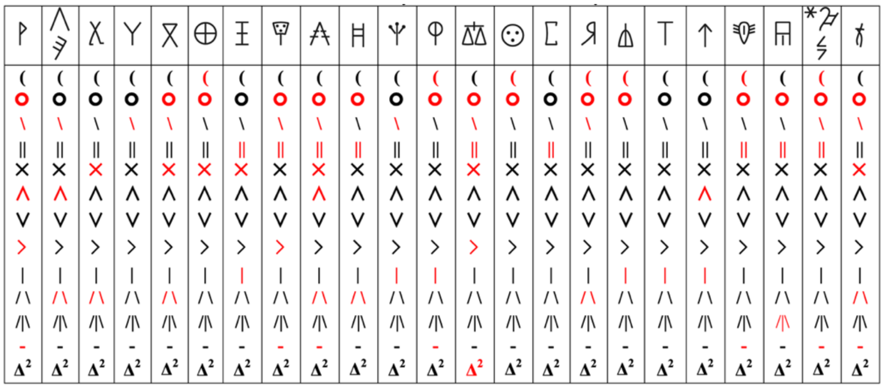


**Figure 1: Feature analysis of Linear A signs according to Revesz [9].**

Sir Alexander Cunningham. Mahadevan [5] proposed a list of signs with 417 distinct symbols in 1977. Later, the Corpus of Indus Seals and Inscriptions (CISI) introduced 386 different symbols [4, 6, 7].

The Indus Valley Civilization originated during the same period as the Sumerian civilization. The Indus Valley and its river tributaries provided basic food and transportation to the people like the Euphrates and the Tigris Rivers in Mesopotamia. The Indus Valley civilization had brick homes, baths, and forts, and used copper and bronze metals to make tools and weaponry. Different seals were used for commerce which were attached to trade goods and showed a mix of symbols. The most important settlement areas were Mohenjo-Daro and Harappa which contained about 35,000 people. Much research showed evidence of trade between Indus Valley and Mesopotamia [12].

The Dravidian language family represents about thirty languages that are common today in Southern India, including the Kannada, Malayalam, Tamil, and Telugu [14]. Daggumati and Revesz [1–3] suggests the possibility of the migration of proto-Dravidian people to the Indus Valley from Mesopotamia because Sumerian pictograms are the most like Indus Valley Script signs among a set of ancient scripts. In addition, Proto-Dravidian *piru* and Mesopotamian *pirus* both mean 'elephant' [12]. The prevalence of Dravidian cognates in the Rig-Veda suggests that Dravidian and Aryan speakers had merged into one language in the large Indo-Gangetic Plain by the time of its composition, while independent Dravidian groups had moved to the boundary of the Indo-Aryan area. The history of Dravidian language evolution is hard to study because the earliest Tamil inscriptions, which were found in the Madurai and Tirunelveli districts of Tamil Nadu, date only from the 2nd century BCE. Perhaps the decipherment of the Indus Valley script could shed more light on the evolution of Dravidian languages.

# 3 A NEW FEATURE ANALYSIS METHOD

In this paper, we follow the third method of finding similarity among scripts, that is, by using feature analysis and similarity matrices.

## 3.1 Feature Analysis

The concept of developing features and thereby presenting the results using similarity matrices is initially suggested by Revesz [8, 9].

Revesz [9] found thirteen features that seem to commonly occur in various scripts. These thirteen features can distinguish all the signs in various ancient scripts. For example, Figure 1 shows a feature analysis of the Minoan Linear A script, where features have a symbol (contains curved line: (, contains an enclosed region: O, has a slanted straight line: etc.). Features that are present are marked as red and features that are absent are marked as black. Given feature tables for two different scripts, a similarity matrix can be generated from them, such as for the Linear A script and the Carian alphabet [2]. In a general view, a similarity matrix helps us to visualize how close the two scripts are at a higher level. This similarity matrix is created by calculating the absolute difference between features of a particular letter in one evaluation table to all the features of a letter in the other evaluation table. This process is to be done for all features of each letter in the first evaluation table. The output of this process will be a distance matrix. Then we need to subtract every element in the distance matrix with total number of features, thirteen in this case, to get the similarity matrix.

## 3.2 Our approach

We have considered the Indus Valley Script and those scripts that are used to write the Dravidian languages of Kannada, Malayalam, Tamil, and Telugu. We applied feature analysis on these languages and try to find similarities among them. We considered 25 of the most common letters from each language and started our process. Unlike western language scripts the Dravidian scripts are more cursive, and we were required to add some extra features to the thirteen features that were proposed in [9]. The new features help to analyze some details of the cursive Dravidian scripts to improve the accuracy of defining the script signs and comparing them. Figure 2 shows the additional features that we introduced for the sake of an improved analysis.

In Figure 2, the check mark has been a predominant feature in the Telugu scripts and has played a major role in changing the pronunciation of the script signs. In the Kannada, Tamil, and Telugu scripts the presence of a short vertical line, dot, and upper curve have a very different meaning were compared to their absence in the signs of these scripts. The horizontal line in the Malayalam script alone distinguishes more than two signs. Finally, we included parallel curves as these Dravidian scripts are more cursive than

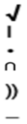


**Figure 2: We introduce the following new features from top to bottom: check mark, short vertical line, dot, upper curve, parallel curves, and horizontal line.**

the straight-line strokes. For example, there are some Telugu script signs that are differentiated with a single dot mark alone.

After developing these feature analysis tables, we needed to create similarity matrices between any two considered language scripts. This Similarity matrix will be a N x N matrix where N is the number of considered letters for the analysis. Hence, each similarity matrix in our context will be 25 x 25 matrix and contain 625 entries. Therefore, calculating all these entries manually is a very time-consuming process besides being prone to mistakes. Hence, we decided to develop a computer program such that it calculates all the values accurately and effectively. Below we present the process of how we treated the values in the feature evaluation table and used them as inputs in the similarity matrix, together with how we developed the logic for the matrix calculation.

Initially we wanted to consider all the features for a particular sign as a single vector. Hence, the features that are marked red (the features which are present in the letter) are considered as 1's and the remaining black marked features (the features which are not present in that letter) are considered as 0's. Therefore, we can extract a total of 25 vectors (from the 25 signs) from one feature evaluation table. These 25 vectors were compared separately with all other 25 feature vectors of the second feature evaluation table. Figure 3 shows the feature analysis matrix for the Malayalam script. Figure 4 shows the feature analysis matrix for the Telugu script.

After the formation of the two feature matrices, we need to transpose one of the matrices to facilitate certain matrix operations. Here we have two 25 x 16 matrices and since we need to perform multiplication functions during the process of forming a similarity matrix, we will encounter a dimensional mismatch error if we do not transpose one of the two feature vector matrices.

We had everything set to apply our main operation to create the similarity matrix, but the question is what this main operation exactly should be. Before discussing that, let's comprehend and analyze how we form a similarity matrix in the traditional way. We calculate the absolute difference between two features in their respective position and remove this difference from the total features value to get the similarity number. For doing this we initially tried with three methods. One is by using the dot product. We all know that the dot product tells us about the angle between the two vectors (A.B = A*B*cos($\theta$)) where $\theta$ is the angle which determines by how much these two vectors got deviated from one another. When we try implementing this model unlike the real dot product the machine was performing a simple matrix multiplication (a weighted sum of vectors) due to which we tend to lose some of the feature values.

In the second method we try implementing XOR operation on the feature vectors which return value 1 only when there are different corresponding vectors (0 and 1, 1 and 0) which exactly what we expect the result to be. But again, we encountered trouble during its implementation. Applying the XOR operation upon the vectors gives the bitwise XOR results rather than the element-wise results. Due to this, the final matrix has a dimension of 25 x 16 unlike the square matrix 25 x 25 that we expect.

The third method is more like a hybrid of the first two methods. It performs Elementwise XOR weighted sum on the vector matrices giving us the absolute difference of a particular feature vector with all feature vectors in the other vector matrix and vice-versa. This result is a 25 x 25 matrix with correct and true values. This generated matrix is a distance matrix and in-order to get the similarity matrix we must subtract every entry in the distance matrix with 16 which is the total features taken for our problem domain. The high value numbers in the similarity matrix represents the strong closeness and low values represent the least connectivity between the corresponding signs in the similarity matrix.

Finally, we presented these similarity matrices using heat maps for better visualization. We used a color gradient from bright blue to dark red to represent the values inside the matrix where red is assigned for high values and blue for low values.

## 4 DISCUSSION

In this section we present the feature analysis for the Malayalam Script, screenshots of our process consisting of different matrices we discussed earlier and finally some output heat maps. The heat maps are presented for the Telugu-Malayalam and Kannada-Telugu languages which contains the total of sixteen features in the feature evaluation table.

In the upper left of Figure 5 from the feature evaluation table all the 16 features for 25 signs are represented in vector notation making it a 25 x 16 matrix, where 25 is the number of signs and 16 is the number of features. Since we need two vector matrices to create a similarity matrix we transpose (upper right of Figure 5) one of the vector matrices to facilitate the elementwise XOR multiplication. The dot product (25 x 25) of these two matrices, and the XOR matrix do not lead us to the similarity matrix because they perform a simple matrix multiplication and bitwise XOR (25 x 16) respectively. To create a similarity matrix, we need to perform Elementwise XOR multiplication (25 x 25) of the matrices, which calculates the weighted sum of absolute difference between any two feature vectors as shown in the lower left of Figure 5. This is the definition of a distance matrix. The similarity matrix is found by subtracting the total number of features with every element in the 25 x 25 distance matrix as shown in the lower right of Figure 5.

From a similarity matrix, it is easy to generate a heat map. For example, the Malayalam-Telugu heat map is hsonw in Figure 6, and the Kannada-Telugu heat map is shown in Figure 7. We can see the highest value of 16 and lowest value of 8 which shows that there are high similar signs and many low similar signs respectively. The graph shows that it is majorly dominated by the red color rather than blue which shows there is a lot of similarity between the two language scripts. Similarly considering the Malayalam and Telugu heat map there are a smaller number of highly matched words which have value of 16 and there is a lot of blue signs in the heat

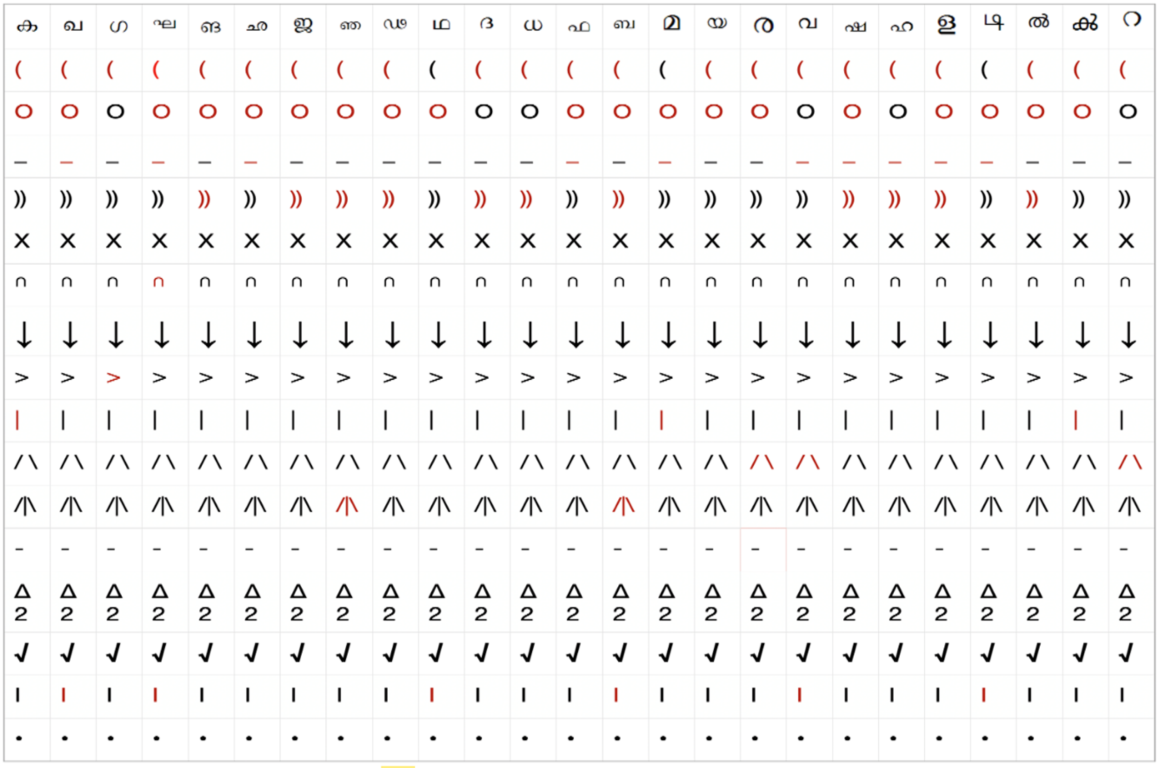


**Figure 3: Feature analysis of the Malayalam script.**

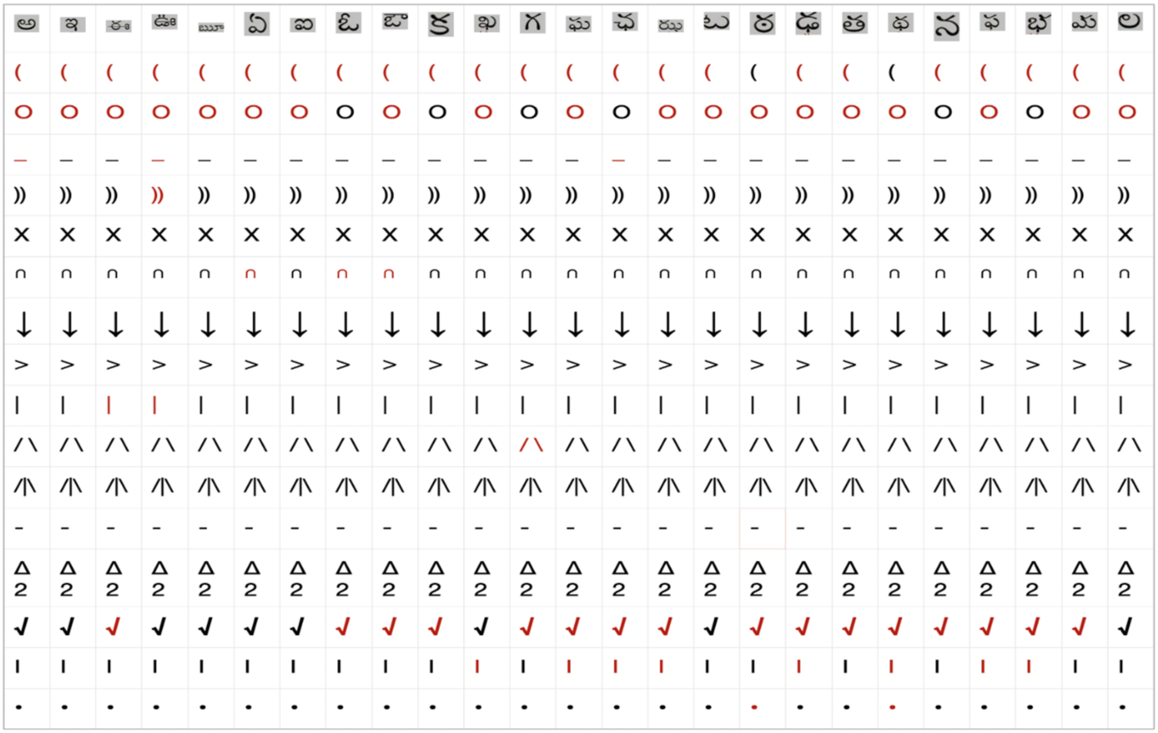
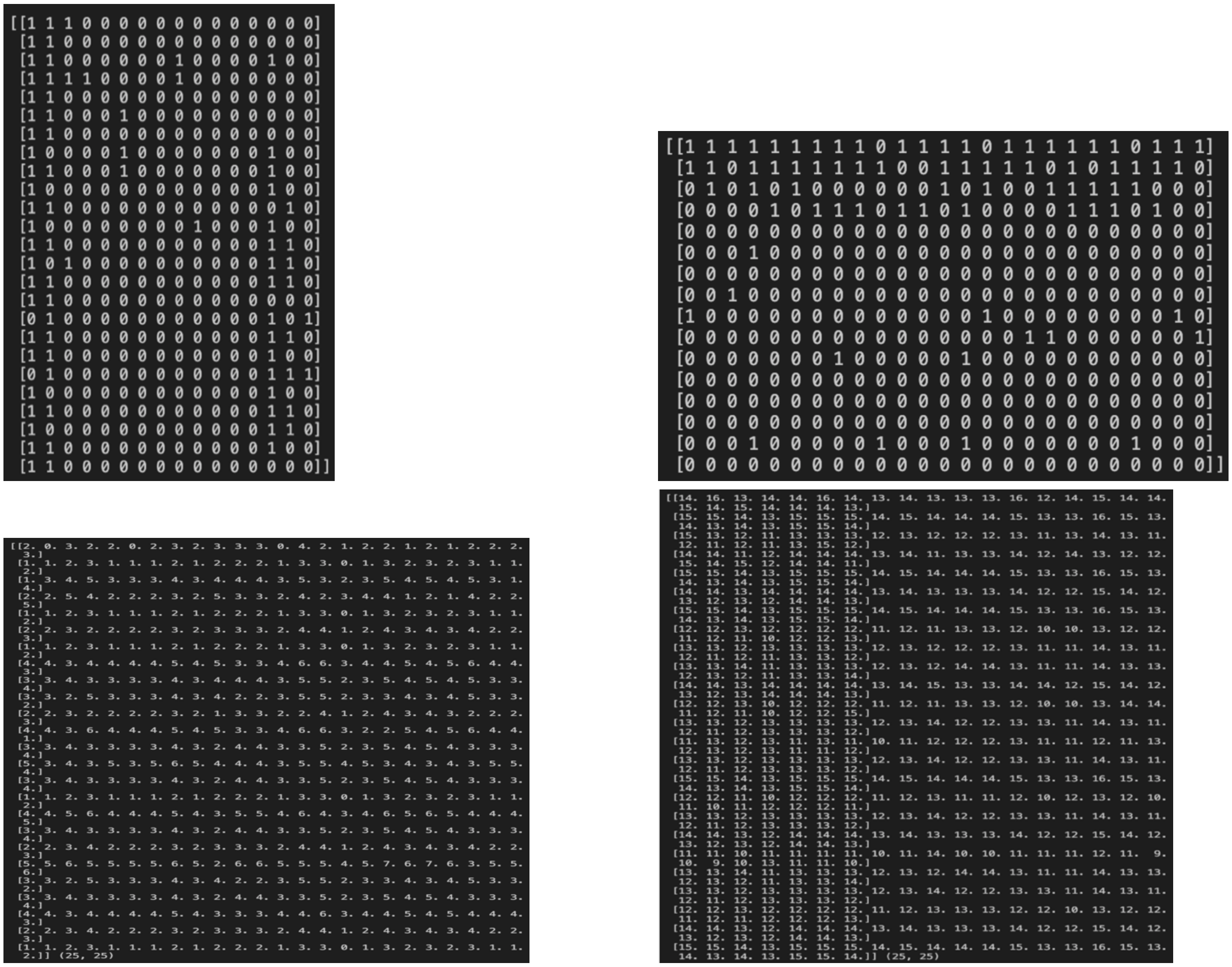


**Figure 4: Feature analysis of the Telugu script.**



```
[[1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0]
 [1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0]
 [1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
 [1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

```
[[1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1]
 [1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 0 1 1 1 1 0]
 [0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0]
 [0 0 0 0 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1]
 [0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

```
[[2. 0. 3. 2. 2. 0. 2. 3. 2. 3. 3. 3. 0. 4. 2. 1. 2. 2. 1. 2. 1. 2. 2. 2.
  3.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]
 [1. 3. 4. 5. 3. 3. 3. 4. 3. 4. 4. 4. 3. 5. 3. 2. 3. 5. 4. 5. 4. 5. 3. 1.
  4.]
 [2. 2. 5. 4. 2. 2. 2. 3. 2. 5. 3. 3. 2. 4. 2. 3. 4. 4. 1. 2. 1. 4. 2. 2.
  5.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]
 [2. 2. 3. 2. 2. 2. 2. 3. 2. 3. 3. 3. 2. 4. 4. 1. 2. 4. 3. 4. 3. 4. 2. 2.
  3.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]
 [4. 4. 3. 4. 4. 4. 4. 5. 4. 5. 3. 3. 4. 6. 6. 3. 4. 4. 5. 4. 5. 6. 4. 4.
  3.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 4. 4. 4. 3. 5. 5. 2. 3. 5. 4. 5. 4. 5. 3. 3.
  4.]
 [3. 3. 2. 5. 3. 3. 3. 4. 3. 4. 2. 2. 3. 5. 5. 2. 3. 3. 4. 3. 4. 5. 3. 3.
  2.]
 [2. 2. 3. 2. 2. 2. 2. 3. 2. 1. 3. 3. 2. 2. 4. 1. 2. 4. 3. 4. 3. 2. 2. 2.
  3.]
 [4. 4. 3. 6. 4. 4. 4. 5. 4. 5. 3. 3. 4. 6. 6. 3. 2. 2. 5. 4. 5. 6. 4. 4.
  1.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 2. 4. 4. 3. 3. 5. 2. 3. 5. 4. 5. 4. 3. 3. 3.
  4.]
 [5. 3. 4. 3. 5. 3. 5. 6. 5. 4. 4. 4. 3. 5. 5. 4. 5. 3. 4. 3. 4. 3. 5. 5.
  4.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 2. 4. 4. 3. 3. 5. 2. 3. 5. 4. 5. 4. 3. 3. 3.
  4.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]
 [4. 4. 5. 6. 4. 4. 4. 5. 4. 3. 5. 5. 4. 6. 4. 3. 4. 6. 5. 6. 5. 4. 4. 4.
  5.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 2. 4. 4. 3. 3. 5. 2. 3. 5. 4. 5. 4. 3. 3. 3.
  4.]
 [2. 2. 3. 4. 2. 2. 2. 3. 2. 3. 3. 3. 2. 4. 4. 1. 2. 4. 3. 4. 3. 4. 2. 2.
  3.]
 [5. 5. 6. 5. 5. 5. 5. 6. 5. 2. 6. 6. 5. 5. 5. 4. 5. 7. 6. 7. 6. 3. 5. 5.
  6.]
 [3. 3. 2. 5. 3. 3. 3. 4. 3. 4. 2. 2. 3. 5. 5. 2. 3. 3. 4. 3. 4. 5. 3. 3.
  2.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 2. 4. 4. 3. 3. 5. 2. 3. 5. 4. 5. 4. 3. 3. 3.
  4.]
 [4. 4. 3. 4. 4. 4. 4. 5. 4. 3. 3. 3. 4. 4. 6. 3. 4. 4. 5. 4. 5. 4. 4. 4.
  3.]
 [2. 2. 3. 4. 2. 2. 2. 3. 2. 3. 3. 3. 2. 4. 4. 1. 2. 4. 3. 4. 3. 4. 2. 2.
  3.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]] (25, 25)
```

```
[[14. 16. 13. 14. 14. 16. 14. 13. 14. 13. 13. 13. 16. 12. 14. 15. 14. 14.
  15. 14. 15. 14. 14. 14. 13.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]
 [15. 13. 12. 11. 13. 13. 13. 12. 13. 12. 12. 12. 13. 11. 13. 14. 13. 11.
  12. 11. 12. 11. 13. 15. 12.]
 [14. 14. 11. 12. 14. 14. 14. 13. 14. 11. 13. 13. 14. 12. 14. 13. 12. 12.
  15. 14. 15. 12. 14. 14. 11.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]
 [14. 14. 13. 14. 14. 14. 14. 13. 14. 13. 13. 13. 14. 12. 12. 15. 14. 12.
  13. 12. 13. 12. 14. 14. 13.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]
 [12. 12. 13. 12. 12. 12. 12. 11. 12. 11. 13. 13. 12. 10. 10. 13. 12. 12.
  11. 12. 11. 10. 12. 12. 13.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 12. 12. 12. 13. 11. 11. 14. 13. 11.
  12. 11. 12. 11. 13. 13. 12.]
 [13. 13. 14. 11. 13. 13. 13. 12. 13. 12. 14. 14. 13. 11. 11. 14. 13. 13.
  12. 13. 12. 11. 13. 13. 14.]
 [14. 14. 13. 14. 14. 14. 14. 13. 14. 15. 13. 13. 14. 14. 12. 15. 14. 12.
  13. 12. 13. 14. 14. 14. 13.]
 [12. 12. 13. 10. 12. 12. 12. 11. 12. 11. 13. 13. 12. 10. 10. 13. 14. 14.
  11. 12. 11. 10. 12. 12. 15.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 14. 12. 12. 13. 13. 11. 14. 13. 11.
  12. 11. 12. 13. 13. 13. 12.]
 [11. 13. 12. 13. 11. 13. 11. 10. 11. 12. 12. 12. 13. 11. 11. 12. 11. 13.
  12. 13. 12. 13. 11. 11. 12.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 14. 12. 12. 13. 13. 11. 14. 13. 11.
  12. 11. 12. 13. 13. 13. 12.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]
 [12. 12. 11. 10. 12. 12. 12. 11. 12. 13. 11. 11. 12. 10. 12. 13. 12. 10.
  11. 10. 11. 12. 12. 12. 11.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 14. 12. 12. 13. 13. 11. 14. 13. 11.
  12. 11. 12. 13. 13. 13. 12.]
 [14. 14. 13. 12. 14. 14. 14. 13. 14. 13. 13. 13. 14. 12. 12. 15. 14. 12.
  13. 12. 13. 12. 14. 14. 13.]
 [11. 11. 10. 11. 11. 11. 11. 10. 11. 14. 10. 10. 11. 11. 11. 12. 11.  9.
  10.  9. 10. 13. 11. 11. 10.]
 [13. 13. 14. 11. 13. 13. 13. 12. 13. 12. 14. 14. 13. 11. 11. 14. 13. 13.
  12. 13. 12. 11. 13. 13. 14.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 14. 12. 12. 13. 13. 11. 14. 13. 11.
  12. 11. 12. 13. 13. 13. 12.]
 [12. 12. 13. 12. 12. 12. 12. 11. 12. 13. 13. 13. 12. 12. 10. 13. 12. 12.
  11. 12. 11. 12. 12. 12. 13.]
 [14. 14. 13. 12. 14. 14. 14. 13. 14. 13. 13. 13. 14. 12. 12. 15. 14. 12.
  13. 12. 13. 12. 14. 14. 13.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]] (25, 25)
```

**Figure 5: Telugu feature matrix (upper left), transpose of the Malayalam feature matrix (upper right), distance matrix (lower left), and a Malayalam-Telugu similarity matrix (lower right).**

map with lowest value of 9. This shows that both scripts differ a lot compared to the above heat map.

## 5 CONCLUSION AND FUTURE WORK

The Dravidian Languages which include Telugu, Tamil, Kannada, and Malayalam are generally known as distinct cousins and are relatively closely related when compared to the Indus Valley Script. Indus valley scripts have been undeciphered until today but there has been a lot of extraction of different kinds of symbols and seals recently. Among the Dravidian languages Telugu and Kannada seem closely related. Though some of the signs in the Tamil script contain a straight-line stroke most of the other signs and signs in other three Dravidian scripts are cursive. This project helps in finding out the similarity between the scripts that are expected to be derived from the undeciphered scripts and help us in finding out the evolution of languages. Our goal is to ease the exhaustive calculations in finding out the similarity matrix between two scripts during comparison. The project has a high scalability factor. It can be extended by passing the feature vector values directly from the created vector table rather than passing them through NumPy arrays. This process can be flexibly applied to words and thereby construct an evolutionary tree as a future work. In addition, feature analysis can be extended from script analysis to art motif analysis [10] and higher-level textual analysis [11].

## REFERENCES

[1] Shruti Daggumati and Peter Z. Revesz. 2018. Data mining ancient script image data using convolutional neural networks. In Proceedings of the 22nd International Database Engineering and Applications Symposium (IDEAS'18), ACM Press, pp. 209-218. https://doi.org/10.1145/3216122.3216163

[2] Shruti Daggumati and Peter Z. Revesz. 2019. Data mining ancient scripts to investigate their relationships and origins. In Proceedings of the 23rd International Database Engineering and Applications Symposium (IDEAS'19), ACM Press, pp. 209-218. https://doi.org/10.1145/3331076.3331116

[3] Shruti Daggumati and Peter Z. Revesz. 2021. A method of identifying allographs in undeciphered scripts and its application to the Indus Valley Script. Humanities and Social Sciences Communications, 8, 50. https://doi.org/10.1057/s41599-021-00713-0

[4] Walter Fairservis, Sayid Ghulam Mustafa Shah, and Asko Parpola. 1993. Corpus of Indus Seals and Inscriptions. Vol. 2: Collections in Pakistan. Journal of the

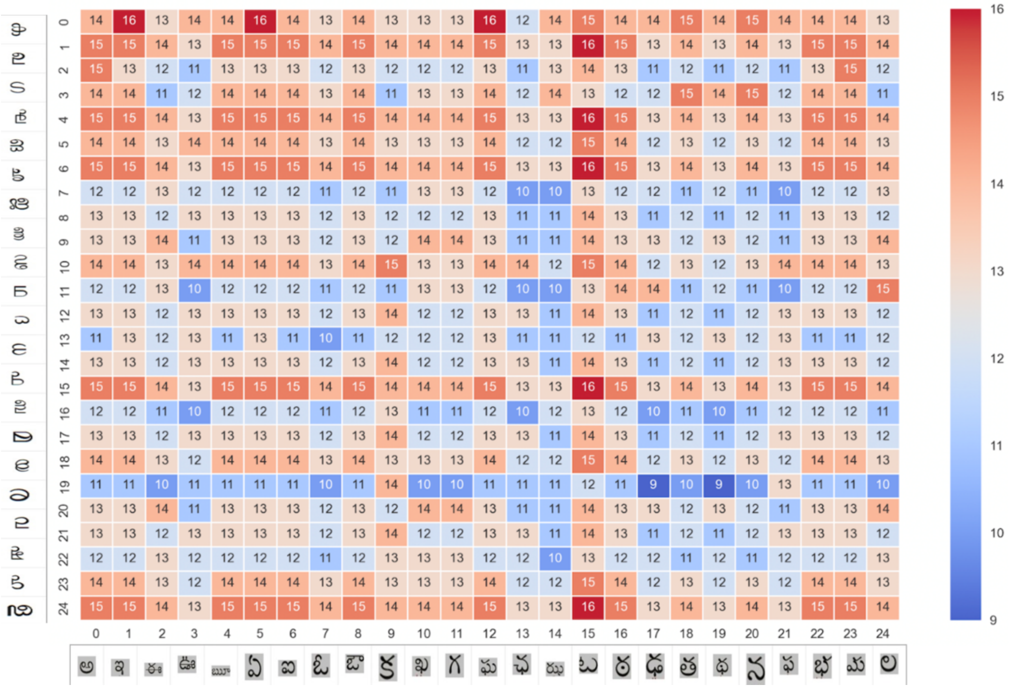


Figure 6: Heat map for Malayalam and Telugu scripts.

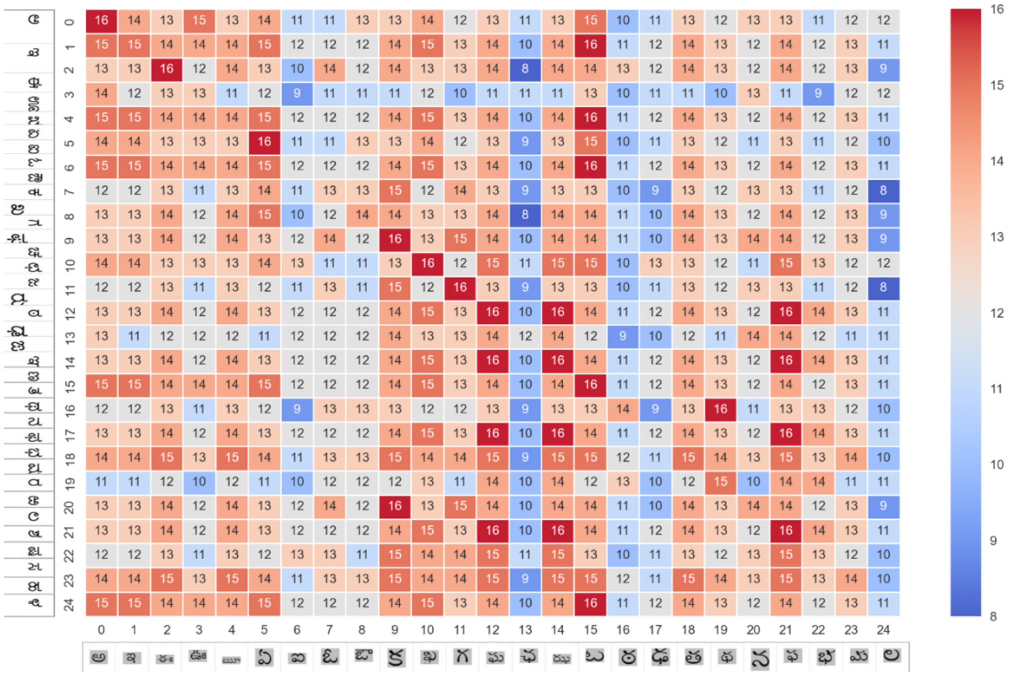


Figure 7: Heat map for Kannada and Telugu scripts.

American Oriental Society, 113 (2), 310.

[5] Iravatham Mahadevan. 1977. The Indus Script: Texts, concordance and tables, memoirs. Archaeological Survey of India, no. 77.

[6] Asko Parpola, Brij Mohan Pande, and Petteri Koskikallio. 2010. Corpus of Indus Seals and Inscriptions. Vol. 3: New material, untraced objects, and collections outside India and Pakistan, Annales Academiae Scientiarum Fennicae, Humaniora; no. 359.

[7] Jagat Pati Joshi and Asko Parpola. 1987. Corpus of Indus Seals and Inscriptions. Vol. 1: Collections in India. Annales Academiae Scientiarum Fennicae. Series B. no. 239.

[8] Peter Z. Revesz, 2016. Bioinformatics evolutionary tree algorithms reveal the history of the Cretan Script Family. International Journal of Applied Mathematics and Informatics, 10, 67-76.

[9] Peter Z. Revesz, 2017. Establishing the West-Ugric language family with Minoan, Hattic and Hungarian by a decipherment of Linear A. WSEAS Transactions on Information Science and Applications, 14, 306-335.

[10] Peter Z. Revesz, 2019. Art motif similarity measure analysis: Fertile Crescent, Old European, Scythian and Hungarian elements in Minoan culture. WSEAS Transactions on Mathematics, 18, 264-287.

[11] Peter Z. Revesz, 2019. A comparative analysis of Hungarian folk songs and Sanskrit literature using motif similarity matrices. WSEAS Transactions on Information Science and Applications, 16, 75-86.

[12] Abumugam Sathasivam. 1965. Sumerian: A Dravidian Language. Berkeley, California.

[13] Maggie Tallerman. 200. Did our ancestors speak a holistic protolanguage? Lingua, 117 (3), 579-604.

[14] Wikipedia, Dravidian languages. 2022. Available at: https://en.wikipedia.org/wiki/Dravidian_languages

# Multi-objective query optimization in Spark SQL

Michail Georgoulakis Misegiannis
School of ECE, NTU Athens
Athens, Greeece
michailgeorgoulakis@dblab.ece.ntua.gr

Verena Kantere*
School of ECE, NTU Athens
Athens, Greece
verena@dblab.ece.ntua.gr

Laurent d'Orazio*
Univ. Rennes, CNRS, IRISA
Lannion, France
laurent.dorazio@univ-rennes1.fr

## ABSTRACT

Query optimization is a challenging process of DBMSs. When tackling query optimization in the cloud, there exists a simultaneous need of providing an optimal physical query execution plan, as well as an optimal resource configuration among available ones. Cloud computing features like resource elasticity and pricing make the process of finding this optimal query plan a multi-objective problem, with the monetary cost being an equally important factor to query execution time. Apache Spark is a popular choice for managing big data in the cloud. However, query optimization in its SQL module (Spark SQL) involves a number of limitations due to the rule-based nature of its optimizer, Catalyst. We propose a multi-objective cost model for the extension of the query optimizer of Apache Spark, aiming to minimize both objectives of query execution time and monetary cost, as well as a methodology for exploring the space of Pareto-optimal query plans and selecting one. The cost model is implemented and tuned, and an experimental study is conducted to validate its accuracy.

## CCS CONCEPTS

• **Information Systems → Query optimization**.

## KEYWORDS

query optimization, cost model, cloud computing, multi-objective optimization, Apache Spark, Catalyst optimizer

**ACM Reference Format:**
Michail Georgoulakis Misegiannis, Verena Kantere, and Laurent d'Orazio*. 2022. Multi-objective query optimization in Spark SQL. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3548785.3548800

## 1 INTRODUCTION

Query optimization is the most challenging step of query processing. A query optimizer can either be rule-based, using heuristics to convert the logical query plan to a physical one, or cost-based, using cost functions to compare alternative query plans and return the optimal according to its estimations. The architecture of a cost-based query optimizer consists of three key stages that determine the quality of its predictions [9]: cardinality estimation, cost modeling and plan enumeration.

The majority of works on query optimization aims to minimize query execution time on fixed hardware, which is a valid assumption in the on-premise world. Query processing in the Cloud, however, presents an extra challenge as alternative hardware instances are available. Depending on the resource configuration that is used, the execution might be completely different. Decision making involves deciding on the type of the cluster, the number of instances that will be used, their type, and their characteristics (e.g. RAM size). In such a scenario, a query optimizer should be able to pick both an optimal physical query execution plan, as well as a resource configuration among available hardware instances, thus bridging the gap between query and resource optimization [18].

In order to achieve this, optimizer cost models should be hardware-agnostic, being able to model the behaviour of a query plan in different clusters and systems. A hardware-agnostic cost model could lead to lower costs as well as better resource efficiency [14].

Query optimization is usually associated with minimizing query execution time. The performance of a query, however, can be evaluated in terms of more objectives. Features of the cloud, like resource elasticity and pricing increase the objectives that can be simultaneously optimized in a cloud setting [5]. Adding instances to achieve maximum parallelization during query execution will in general lead to lower execution times, but may also lead to much higher monetary costs, as well as increased energy consumption. Monetary cost is one of the most prevalent query optimization objectives in the cloud [8, 10, 15]. Energy consumption has also been considered lately [13], as cloud providers yearn for reducing energy cost. Other objectives that can be considered in multi-objective query optimization are result precision [16], or data security.

Query optimization in big data systems, which are usually hosted in the cloud, is particularly challenging. As a result, it is necessary that the estimation components of query optimizers (cardinality estimator, cost model) are accurate. One of the most popular big data processing frameworks is Apache Spark, which is widely used in research and industry. However, the query optimizer of Spark's SQL-based component, Spark SQL, has a limited cost model.

In this work, we propose a multi-objective cost model for Spark SQL, for the objectives of query execution time and monetary cost. For the time objective, we adopt an existing single objective cost model [4] for Spark SQL, which shows promising accuracy. We also conduct a detailed experimental study to validate it. For the money objective, we introduce a formula in order to estimate the monetary cost of a query, based on real cloud pricings.

[*] Both supervisors contributed equally to this work.

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548800

The cost model receives a query and a set of Spark application configurations as an input, and returns the optimal query plans for each configuration, resulting in a Pareto front. The returned plans present different tradeoffs for the two objectives, and the user can either select one or assign preference weights to the two objectives, and be provided with a single query plan that best meets them.

We also conduct an experimental study on a private cloud environment to validate the cost model accuracy and optimality for a broadly adopted architecture, consisting of Spark, Yarn and HDFS.

Overall, the contributions of this work are the following:

- proposal of a multi-objective cost model for Spark SQL
- introduction of a formula for query monetary cost estimation
- reimplementation and validation of existing cost model for query execution time estimation
- a detailed experimental study on a real private cloud environment
- a user-interactive method for exploring the space of alternative query plans and choosing an optimal one

The rest of this paper is organized as follows. Related work is discussed in Section 2. Section 3 describes the cost model that we implemented. The experimental evaluation and its results follow in Section 4, while Section 5 concludes the paper.

## 2 RELATED WORK

**Optimization in Spark** The Spark SQL query optimizer, Catalyst [3], is an extensible optimizer where new rules can be added. However, it is not ideal for cost-based query optimization. It only uses a limited cost model, being unable to provide analytical estimations for the execution time of a query plan. A number of research works have focused on improving specific limitations of the Catalyst optimizer [17].

Although Spark is highly configurable, its manual tuning is time consuming and complex, due to the high-dimensional configuration space. A lot of works provide frameworks for tuning Spark applications [12, 15], in most cases with learned methods. The proposed cost model can be useful in this perspective too, as apart from producing optimal query plans, it can also be used for tuning and comparing different application configurations.

**Multi-objective query optimization** Karampaglis et al. [6] proposed a bi-objective query cost model suitable for query optimization over a multi-cloud environment. It successfully provides estimates of both the expected execution time and monetary cost.

A number of works have considered multi-objective query optimization in the cloud. Kllapi et al. [7] proposed a technique to optimize dataflow scheduling on a set of containers and form one schedule best meeting user constraints. Their work can also be used for query optimization, when the execution of a query can happen over multiple containers. Multi-objective parametric query optimization [16] takes a different approach to query optimization, which happens before runtime with the use of an exhaustive DP algorithm, and models queries as functions of parameters.

## 3 COST MODEL AND IMPLEMENTATION

In this work, we propose a cost model for cost-based multi-objective query optimization in Apache Spark. For the objective of query execution time, we adopt a proposed cost model for Spark SQL [4], which we also experimentally evaluated. For the objective of monetary cost, we introduced a cost estimation formula for a given query plan. By combining them, we tackle query optimization as a multi-objective optimization (MOO) problem and use different methods to explore the space of Pareto-optimal query plans.

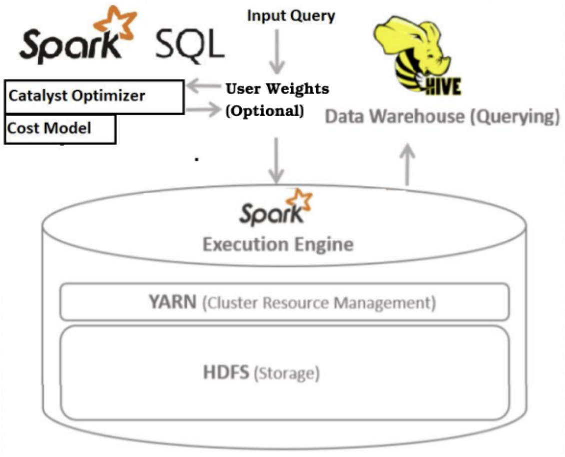


**Figure 1: System architecture**

| | Catalyst C.M. | Proposed C.M. |
|---|---|---|
| Query types | ALL | GPSJ |
| Cost based join selection | YES | YES |
| Tables and Columns statistics | YES | YES |
| Considers cluster topology | NO | YES |
| Based on system disk access time | NO | YES |
| Takes into account network speed | NO | YES |
| Analytic estimation of QET | NO | YES |

**Table 1: Comparison of Catalyst and proposed cost model**

### 3.1 System Architecture

Figure 1 shows the system upon which the cost model operates. The storage layer includes a number of datanodes inside an HDFS filesystem. Data is processed in Spark, and Yarn is the resource negotiator between HDFS and Spark. For data management, data is stored in Apache Hive tables and accessed through Spark SQL queries. As for the optimizer, an extended version of the Catalyst is envisioned, operating cost-based by using the proposed cost model. We implemented the cost model outside Spark and used it manually.

### 3.2 Cost Model Preliminaries

The proposed single-objective Spark SQL cost model [4] can provide more than Catalyst when it comes to estimating query execution, as highlighted by Table 1. It is based on disk access time and network speed, as disk and network performance is critical in one-pass workloads, like Spark SQL queries. It is a reconfigurable model, that can be tuned for any (homogeneous) cluster and system. It also performs traditional SQL optimizations by collecting table and columns statistics. It is aware of the cluster topology, and takes into account Spark application parameters that influence query execution time. The Spark application parameters considered are the number of Spark executors, and the number of executor cores.

One of its limitations is that it covers the class of Generalized Projection, Selection, and Join (GPSJ) queries, which are a subset of

SQL queries. This means that the use of specific SQL operators like UNION ALL or OUTER JOIN are not supported by the cost model. In addition to that, its use is also limited for homogeneous clusters.

The cost model is capable of analytically estimating the execution time of the five essential RDD transformations that occur in Spark SQL GPSJ queries: (1) Full table scan (2) Full table scan and broadcast (3) Shuffle hash join (4) Broadcast hash join (5) Group by. Each of the transformations is modeled as a function in the cost model code. Precisely modeling these operations is challenging, so the cost model focuses on a set of basic bricks that determine transformations and actions cost, for which it provides cost estimates (Read, Write, Shuffle Read, Broadcast). Each one of these functions receives a data table set as an input (or two, in case of a join operator), as well as the table's cardinality, size, partitions and any filtering predicates. It returns the estimated time needed to execute the transformation, the columns returned, the cardinality and the adjusted size of the table set. The estimated execution time for a query is obtained by summing up the time needed to execute each RDD transformation forming the query physical plan.

### 3.3 Bi-objective cost model

In the Spark-Yarn-HDFS architecture, query execution involves two parts. A user submits a query, and then specifies some parameters to configure the Spark application. Spark application tuning, although often done empirically, is a complex decision to make, as Spark has a considerable number of parameters that can be configured. One of the most critical parameters is the number of executors that will be allocated for an application. Each Spark executor runs within a Yarn container. Yarn containers are provided by Yarn on demand at the start of each Spark Application. Each one hosts a Spark executor as well as a number of cores that are assigned to it. They are deallocated when the Spark application completes.

The second part of our cost model involves the prediction of the monetary cost of executing a query. In a public IaaS cloud platform, execution of a Spark SQL query requires renting a number of computing instances to host the Spark executors, using them during the runtime of a query and leasing them when execution is completed. As a result, we make monetary cost estimations with the following formula:

$$cost = \ ci(\#executors) * runtime * hcost(\$/hour)/3600 \qquad (1)$$

ci represents the number of computing instances rented as a function of the number of executors and hcost is the hourly cost of using a single computing instance, which we divide by 3600 to scalarize it to seconds. The formula assumes per-second billing.

To define the ci function, we need to assume a Spark application deployment method. In our work, Spark application deployment was well spread, as we assigned each executor on a different computing instance, in order to achieve maximum parallelization. As a result, $ci(\#executors) = \#executors$

In our experimental part, we used prices from Amazon EC2 instances. For an example of a Spark application with 4 executors and 4 executor cores, we rent 4 homogeneous computing instances with 4vCPUs each, for the time needed for the query to execute. If we use a1.xlarge instances ($0.102 hourly cost) and the query takes 20 minutes to complete, its cost is estimated to be about $0.136.

### 3.4 Query plan enumeration

For a given query, the cost model is able to compare all valid query plans. In our case, two alternative join operators are available and we considere all their combinations, as well as all possible join orderings. We base our monetary cost estimations on the price of renting an a1.medium instance from Amazon EC2, considering cases of renting from one to eight a1.medium instances for query execution. The cost model is easily extensible therefore prices for more computing instance types can be included, to compare different cluster and application scenarios. As a result, our search space for a query involving X join operations involves $2^X$ join operator combinations, $X!$ join orderings, and 8*N available application configurations, for N computing instance types. For example, for TPC-H Query 3 that involves 3 join operations and considering only one computing instance, results in 384 alternative query plans. The search space can become much larger for more complex queries, however our cost model is able to compare all plans for queries with up to 6 joins with no significant optimization overhead.

For the case of even more complex queries, the search space can be reduced significantly if we do not perform join reordering but keep the join order that the Catalyst produces, and if we introduce distributed query optimization heuristics for join selection [11].

In the experimental part, we also had to overcome the limitation of Catalyst returning a single query plan and not providing alternatives. By reconfiguring certain configuration parameters (disabling broadcast joins, changing broadcast joins thresholds), we were able to produce and compare more query plans.

### 3.5 Multi-objective optimization

We follow a three-step process for multi-objective optimization.

**Step 1** First, we apply single-objective optimization to find the optimal query plan in terms of execution time, for every available system configuration. A single query plan is returned for each configuration. As query monetary cost is dependent on execution time, the fastest plan is also the cheapest, for a fixed system configuration, which explains the reason behind this first step. The different tradeoffs are created by the alternative configurations, and not by alternative query plans inside a certain application setting.

**Step 2** The second step is the multi-objective optimization one, as all the query plans from the first step are compared in terms of both objectives. Dominated plans are discarded, and the remaining, Pareto-optimal plans are the output, forming a Pareto front. The selected query plan will determine both the physical query plan that will be executed, as well as the hardware configuration.

**Step 3** After the Pareto front is formed, the final step is the query plan selection. As the number of alternative query plans can be large, the process of presenting the alternatives to the user, and assisting him/her to make a decision is challenging. In order to reduce the number of alternatives presented to the user and take into account budget and needs, price and latency filters can be applied. The cost model can also be used in a user-interactive mode, where the user submits preference weights to the objectives and receives a single plan best meeting them. In that case the problem is scalarized to a single-objective one, using the equation:

$$F(x) = \frac{1}{1 + w_1 * f_1(x)} * \frac{1}{1 + c * w_2 * f_2(x)} \qquad (2)$$

In order to normalize the values of time and money to the same order of magnitude we use constant c. We set c to an empirical value of 25000, so that 25 seconds of query execution are equivalent with a monetary cost of 0.001 US $. The value for c was selected empirically based on our experiments, in which execution time varied between 50-400 seconds depending on the query and configuration and monetary cost between 0.001 - 0.01 US $ per query. In the case of equal weights, a query plan near the middle of the Pareto curve is selected, meaning that it does not prioritize any of the time-money objectives over the other. The query plan that has the maximum value for F is selected for execution. Before the user submits the preferred weights, he/she can also be provided with a value showing the time-money relationship for the selected weights.

## 4 EXPERIMENTAL EVALUATION

**Methodology** The single-objective cost model makes the following assumptions, which our work inherits too:

- It covers the class of GPSJ queries
- It assumes uniform distribution of data in table sets
- It performs single query optimization, assuming a cold start
- It assumes operating on a homogeneous cluster

We evaluated the cost model for a main-memory scenario, assuming that all data fits in memory, as well as intermediate results. We also assumed exclusion of exogenous factors potentially affecting cluster performance, and we calculated a Spark-Yarn initialization overhead in each query, which we did not take into account.

For the evaluation of the cost model, we follow a two step methodology. First, we examine the estimation accuracy of the cost model. Second, we evaluate the optimality of the cost model, aiming to see if it can point to an optimal query plan among alternatives.

The cost model can be used to produce a Pareto front, including plans that offer different time-money tradeoffs. The decision maker can choose the query plan that best suits his/her needs or can use the cost model in a user-interactive mode, and assign weights to the objectives, in order to be provided with a single query plan.

**Setup** We conducted our experiments in Grid '5000 [1], a large-scale and flexible platform for experiment-driven research in computer science, with a focus on parallel and distributed computing.

In our experiments Grid '5000 was used as a private cloud, where we deployed up to 8 homogeneous computing nodes, each one containing 2 CPUs Intel Xeon E5-2630 v3, 8 cores/CPU, 128GB RAM, 5x558GB HDD, 186GB SSD, and 2 x 10Gb Ethernet. Inside our cluster, we set up an HDFS filesystem where each node worked as a datanode, and our dataset was stored in the SSDs.

**Experimental Evaluation** The single-objective cost model was reimplemented to model each one of the five RDD transformations. For our experiments, we used TPC-H benchmark queries and its dataset, scaled to 100GBs. The performance of the cost model was validated for many different factors. This allowed us to fine-tune the cost model for our system, and also re-evaluate it and observe its strengths and inaccuracies. Figure 2 shows its performance for a number of TPC-H queries (7.7% error) in a scenario with 4 Spark executors and 4 executor cores. The estimations are quite accurate with the exception of Query 10, where the execution time of a number of broadcast joins is overestimated. Figure 3 shows that the cost model is also able to capture the impact adding Spark executors has for query execution, with an error rate of 12.1%. The query execution time values are an average for a set of TPC-H queries. A more detailed evaluation of the cost model estimation accuracy can be found in the thesis of Georgoulakis Misegiannis (2021), upon which this paper is based [2].

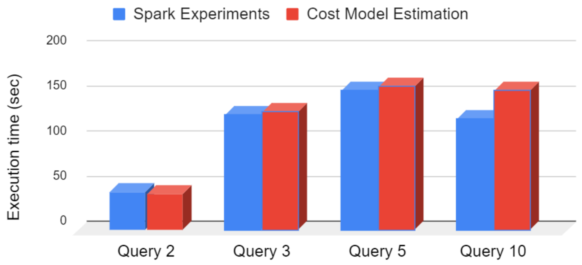


Figure 2: Execution time for different TPC-H queries

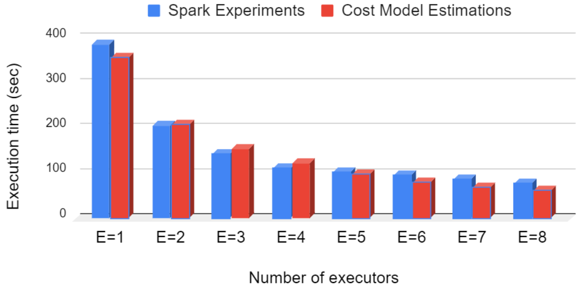


Figure 3: Query Execution times for different number of Spark executors

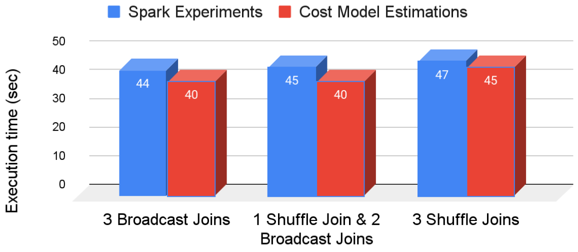


Figure 4: Execution times for three alternative TPC-H Query 2 query plans

Inaccuracies mainly have to do with the fact that the cost model does not precisely model Spark data actions and transformations, but only provides estimates on key operations. Furthermore, in some cases it assumes linear relations between different factors not considering heuristics or overkills that occur in Spark. Finally, stochastic processes happening in the cluster might be influencing system characteristics like the read/write throughput or the network speed, causing minor inaccuracies. As a result, tuning the cost model for a given cluster was a challenging task.

In terms of optimality (prediction accuracy), the cost model makes correct predictions for trivial cases. For complex queries, when it did not point to the optimal plan, it was able to at least spot the trends and propose a near-optimal query plan, having small impact in query execution time. Figure 4 shows the performance of the cost model on 3 alternative query plans for TPC-H Query 2 and a configuration involving 4 Spark executors and 4 executor

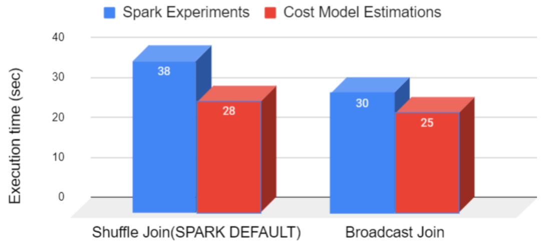


**Figure 5: The cost model corrects the Catalyst**

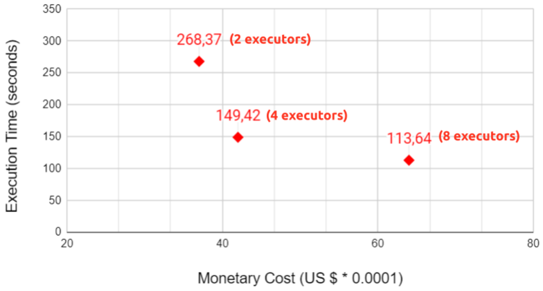


**Figure 6: Pareto Front of the optimal query plans**

cores. As the table sets considered in Query 2 are quite small, the query plan involving only broadcast joins performs better than the alternatives. Shuffle joining in every case results in a slightly worse execution time, whereas operating the first join with a shuffle join operator and the other two with a broadcast one has similar performance with the optimal case. The cost model gives equally good estimates for the two best execution plans.

The cost model is able to make better choices than the Spark SQL optimizer in many cases. For example, for a simple query involving just a join operation (Fig. 5), we can see that a broadcast hash join is a better choice, resulting in 8 seconds faster query execution than using a shuffle hash join. However, the Catalyst by default chooses to shuffle join these tables. The cost model is able to point to the broadcast hash join as the better option. In the experimental study, the cost model proved that it can be a "relevant first step for turning Catalyst into a fully cost based optimizer", showing significant estimation accuracy while staying on point when it comes to optimality and prediction quality.

When it comes to performing multi-objective optimization, Figure 6 shows the outcome of the extended cost model for a scenario with 3 alternative Spark application configurations with a varying number of executors (2,4 and 8), and 4 executor cores.

The cost model returns the fastest query plan for each application configuration, and then the plans are compared in terms of both objectives. In that case, all three query plans are Pareto optimal and form a Pareto front, as they present different time-money tradeoffs. Thus, the problem is formulated to a multi-objective optimization one. The user can either decide himself/herself which query plan best fits his/her application, or can assign weights of preferences to the objectives in the user-interactive mode of the cost model. For the case of $w_1 = w_2$, the plan with the 4 executors is picked. In case of $w_1 = 2 * w_2$, the plan with the 8 executors is picked, and in case of $w_2 = 2 * w_1$ the selected plan is the one with 2 executors, as the time-money relationship changes each time.

## 5 CONCLUSION - FUTURE WORK

In this paper we proposed a multi-objective cost model for query optimization in Spark SQL. We built on a promising proposed cost model, which we extended with a formula for estimating the monetary cost of query plans in Spark. The cost model is able to compare query plans providing different time-money tradeoffs, and we also introduce a method for assisting the user into picking a single one. The cost model was implemented and tested, as a detailed experimental study was conducted in a private cloud environment.

In the future, we aim to extend the cost model to consider heterogeneous resources. This will require modeling the execution costs on different hardware, like GPUs. We also aim to explore more optimization goals, like energy consumption, which is a critical objective in green and sustainable data centers. Finally, a long term goal is to try to integrate the cost model into the Catalyst.

## ACKNOWLEDGMENTS

Experiments presented in this work were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

## REFERENCES

[1] 2005. *Grid'5000.* https://www.grid5000.fr/w/Grid5000:Home
[2] 2021. *Multi-objective query optimization for massively parallel processing in Cloud Computing.* https://dspace.lib.ntua.gr/xmlui/handle/123456789/55115
[3] Michael Armbrust, Reynold Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, and Matei A. Zaharia. 2015. Spark SQL: Relational Data Processing in Spark. *ACM SIGMOD* (2015).
[4] Lorenzo Baldacci and Matteo Golfarelli. 2019. A Cost Model for SPARK SQL. *IEEE Transactions on Knowledge and Data Engineering* 31 (2019), 819–832.
[5] Michail Georgoulakis Misegiannis, Laurent d'Orazio, and Verena Kantere. 2022. From Cloud to Serverless : MOO in the new Cloud epoch. *EDBT* 2022, 1–4.
[6] Zisis Karampaglis, Anastasios Gounaris, and Yannis Manolopoulos. 2014. A Bi-objective cost model for database queries in a multi-cloud environment. *MEDES* (2014), 109–116.
[7] Herald Kllapi, Eva Sitaridi, Manolis M. Tsangaris, and Yannis Ioannidis. 2011. Schedule optimization for data processing flows on the cloud. In *ACM SIGMOD.*
[8] Trung Dung Le, Verena Kantere, and Laurent D'Orazio. 2020. Dynamic Estimation and Grid Partitioning Approach for Multi-objective Optimization Problems in Medical Cloud Federations. *LNCS* 12410 (2020), 32–66.
[9] Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter A. Boncz, Alfons Kemper, and Thomas Neumann. 2015. How Good Are Query Optimizers, Really? *Proc. VLDB Endow.* 9 (2015), 204–215.
[10] Viktor Leis and Maximilian Kuschewski. 2021. Towards Cost-Optimal Query Processing in the Cloud. *Proc. VLDB Endow.* 14, 9 (2021), 1606–1612.
[11] Sunita Mahajan. 2012. General Framework for Optimization of Distributed Queries. *IJDMS* 4 (06 2012), 35–47.
[12] Dimitra Nikitopoulou, Dimosthenis Masouros, Sotirios Xydis, and Dimitrios Soudris. 2021. Performance Analysis and Auto-tuning for SPARK in-memory analytics. *DATE* 2021-Febru, 825061 (2021), 76–81.
[13] A. Sathya Sofia and P. GaneshKumar. 2018. Multi-objective Task Scheduling to Minimize Energy Consumption and Makespan of Cloud Computing Using NSGA-II. *Journal of Network and Systems Management* 26, 2 (2018), 463–485.
[14] Tarique Siddiqui, Alekh Jindal, Shi Qiao, Hiren Patel, and Wangchao Le. 2020. Cost Models for Big Data Query Processing: Learning, Retrofitting, and Our Findings. *ACM SIGMOD* 1 (2020), 99–113.
[15] Fei Song, Khaled Zaouk, Chenghao Lyu, Arnab Sinha, Qi Fan, Yanlei Diao, and Prashant Shenoy. 2021. Spark-based Cloud Data Analytics using Multi-Objective Optimization. (2021), 396–407.
[16] Immanuel Trummer and Christoph Koch. 2017. Multi-objective parametric query optimization. *Commun. ACM* 60, 10 (2017), 81–89.
[17] Alexandru Uta, Bogdan Ghit, Ankur Dave, and Peter Boncz. 2019. [Demo] Low-latency spark queries on updatable data. *ACM SIGMOD* (2019), 2009–2012.
[18] Lalitha Viswanathan, Alekh Jindal, and Konstantinos Karanasos. 2018. Query and Resource Optimization: Bridging the Gap. *ICDE* (2018), 1384–1387.

# A Formal Framework for Data Lakes Based on Category Theory

Alexis Guyot
alexis.guyot@u-bourgogne.fr
LIB Univ. Bourgogne Franche Comté
Dijon, France

Annabelle Gillet
annabelle.gillet@u-bourgogne.fr
LIB Univ. Bourgogne Franche Comté
Dijon, France

Éric Leclercq
eric.leclercq@u-bourgogne.fr
LIB Univ. Bourgogne Franche Comté
Dijon, France

Nadine Cullot
nadine.cullot@u-bourgogne.fr
LIB Univ. Bourgogne Franche Comté
Dijon, France

## ABSTRACT

The management of Big Data requires flexible systems to handle the heterogeneity of data models as well as the complexity of analytical workflows. Traditional systems like data warehouses have reached their limits due to their rigid schema-on-write paradigm, that requires well identified and defined use cases to ingest data. Data lakes, with their schema-on-read paradigm, have been proposed as more flexible systems in which raw data are directly stored in their original format associated with metadata, to be accessed and transformed only when users need to process or analyze them. Thus, it is necessary to define and control the different levels of abstraction and the dependencies among functionalities of a data lake to use it efficiently. In this article, we present a formal framework aiming to define a data lake pattern and to unify the interactions among the functionalities. We use the category theory as theoretical foundations to benefit from its high level of abstraction and its compositionality. By relying on different categories and functors, we ensure the navigation among the functionalities and allow the composition of multiples operations, while keeping track of the entire lineage of data. We also show how our framework can be applied on a simple example of data lake.

## CCS CONCEPTS

• **Information systems** → *Decision support systems*; *Business intelligence*; **Data management systems**; **Information storage systems**; • **Software and its engineering** → *Architecture description languages*; • **Computer systems organization** → *Architectures*;

## KEYWORDS

Data Lakes, Category Theory, Architecture Pattern

**ACM Reference Format:**
Alexis Guyot, Annabelle Gillet, Éric Leclercq, and Nadine Cullot. 2022. A Formal Framework for Data Lakes Based on Category Theory. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3548785.3548797

## 1 INTRODUCTION

Changes in the way of producing and consuming data have led to the emergence of new issues for information systems. Big Data are especially tied to 5 main concerns, usually referred to 5V: Volume, Variety, Velocity, Veracity and Value. Management of such data requires appropriate environments offering enough flexibility to support heterogeneous data with different models, but also multiple data analysis tools and complex pipelines or workflows. Traditional systems like data warehouses have thereby reached their limits due to their schema-on-write paradigm. Indeed, when using Extract-Transform-Load (ETL) processes to ingest data, the time-consuming and difficult data integration and cleaning steps require well known use cases, that hinder the flexibility of such systems.

Data lakes have been proposed as flexible environments for storing and analyzing Big Data [13] with a schema-on-read paradigm. In other words, raw data are usually directly stored in their original format and only processed when needed, but at the cost of complex model transformations that are supported by users. Different kind of metadata are proposed to help users to navigate, select, validate and transform data according to their needs. To a certain extent, it prevents from turning data lakes into data swamps [20], whose usability is reduced because of the difficulty to localize data.

Data lakes have been described multiple times in the literature [9, 41, 44], but we follow Hai et al.'s definition [25]:

> A data lake is a flexible, scalable data storage and management system, which ingests and stores raw data from heterogeneous sources in their original format, and provides query processing and data analytics in an on-the-fly manner.

This definition is fairly complete. It presents the data lake as an integrated system from the user's perspective (for example a data engineer), exposing multiple functionalities. At a technical level, a data lake is in fact an architecture that brings together specialized components.

Despite being defined as systems for both storing and processing data with management and analytical processes, few architectures of data lakes take all these aspects into consideration in a unified

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548797

manner [44]. This has led to the creation of alternative systems like Delta Lakes [2], Lakehouses [3] and Data Mesh [10], that focus on improving a subset of functionalities of data lakes. As a result, these systems have major drawbacks: they do not necessarily meet the functional requirements and they lack robustness. One of the main causes is that they are built as an assembly of isolated components having different behavioral properties, and thus it is difficult to ensure the validity of the expected properties for the whole system.

However, the use of multiple components to support all the functionalities of the data lake is inevitable. Thus, the notion of data lake is in fact an architecture pattern in which the functionalities are well-defined. To avoid the data lake construction issues, some works narrow their system for a specific use case according to different domains [31, 34, 36, 40, 43]. We adopt a more abstract point of view, and aim to define a framework allowing to generalize the data lake pattern and to unify the component interactions.

The greater control offered by robust systems is usually obtained through solid theoretical foundations [5, 29], such as Codd's relational model [8] and algebra for database management systems, which can provide formal frameworks for studying different properties and their preservation or for building optimization processes. Data lakes, as pragmatic solutions originally designed to resolve industrial issues, were not defined at the time with strong theoretical foundations to describe and validate their functionalities, or to control their uses. Nevertheless, they could benefit from such framework.

Category theory [15] is a meta-mathematical formalism introduced in the 1940s by Saunders MacLane and Samuel Eilenberg. It has already been successfully used for building formal frameworks in various domains of computer science like functional programming or software architectures. These domains especially benefit from the high level of abstraction and compositionality of this theory. Data lakes can greatly take advantage of these characteristics, as data and functions need to be represented altogether. Indeed, the schema-on-read paradigm requires enough expressivity to allow all kind of processing, but it also necessitates constraints in order to control data and metadata organization as well as their transformations and analysis. Category theory can provide all of these requirements.

In this article, we propose to use category theory to build a formal framework allowing the interconnections among the different functionalities of a data lake, and unifying the levels of abstraction. It allows to compose the functionalities, and thus to keep track of the lineage of data, in order to give a formal structure to data lakes while coping with their need for flexibility. We also show the usability of the framework by applying it to an example of simple data lake.

The remainder of this paper is structured as follows. First, we describe in section 2 the main functionalities of data lakes according to the literature, we present other formalisms previously proposed for data lakes and we show how the category theory can contribute to the formalization of these systems. In section 3, we introduce the main concepts of the category theory and our formal framework. We then use it to model a small example of data lake in section 3.3. Finally, we draw conclusions of our work and open up perspectives for the future in section 4.

# 2 STATE OF THE ART AND DISCUSSION

In this section, after describing the main functionalities of data lakes, we argue that various levels of abstraction and dependencies among functionalities justify the need of a formalism allowing both abstraction and composition. In a second part, we study the major works regarding the formalization of data lake components and we show how category theory can be used in this context.

## 2.1 Main Functionalities of Data Lakes

Reviews of the literature [9, 25, 41, 44] agree on 4 main functionalities for data lakes, namely Data Storage, Data Ingestion, Data Maintenance and Data Exploration.

**Data Storage** can take several forms in data lakes, from single systems handling data heterogeneity with generic models [16, 39] to polystores built as an assembly of specialized database management systems [4, 24, 27]. As datasets must be properly and accurately annotated with metadata so that the data lake does not become a data swamp [20], the data storage functionality also includes the storage of metadata.

**Data Ingestion** provides tools for connecting the system to data sources, loading the data in streaming and/or batch manner and retrieving or producing basic metadata [19, 38]. Some data may require to only store aggregated insights to reduce their important volume, as it is the case for data streams of sensor data with high velocity.

**Data Maintenance** ensures: 1) the usability of the data by organizing the lake [1, 33] and by extracting more advanced metadata [4, 17, 27], for example through profiling or through the discovery of relationships among datasets; 2) the quality of data [16, 26], by guarantying or improving it, for example through the application of integrity constraints; and 3) the ease of use of the system by providing functionalities that make schema-on-read simpler and more efficient such as schema pre-integration [24].

Finally, **Data Exploration** functionality allows the discovery of content in data lakes through unified query interfaces [4, 24] or navigational algorithms based on measures of relatedness [17] or faceted search [27]. During the exploration phase, datasets are retrieved and integrated in an on-the-fly manner. Queries and algorithms can be applied on them in order to obtain results according to the case study.

The description of the functionalities extracted from the literature reveals two major characteristics of data lakes. Firstly, it brings to light **different levels of abstraction** related to: 1) data, including various models, formats and metadata; 2) software architectures, with different strategies and components that can be used to store and process data; and finally 3) functionalities themselves, composed to implement other higher level services. Secondly, the definitions also reveal existing **dependencies among the main functionalities**. Data exploration depends on data maintenance and on data storage, data storage depends on data ingestion and finally data maintenance depends on data storage.

The higher complexity induced by abstractions and dependencies is a strong motivation for building a formal framework able to ensure the robustness of data lakes and to control the interactions among components.

## 2.2 On Formalization of Data Lakes

Only few proposals have been made to provide such formal framework for data lakes. Most of previous works have focused on the formalization of mostly isolated aspects of data lakes such as metadata models, data storage, analytical queries, etc.

Several models have been proposed for metadata modeling using UML, entity relationship or graph theory. In [45] the authors state that existing metadata models are either tailored for a specific use case or not generic enough to be used in different contexts. They extend their previous model called MEDAL (MEtadata model for DAta Lakes) to build a more generic one called goldMEDAL, defined on three levels: conceptual, logical and physical. The conceptual level is formalized through set theory and describes data entities and groupings, as well as hierarchy and lineage relationships. The logical level puts together the previous elements through graph theory concepts, especially nodes, edges and hyperedges. The physical level is finally implemented with the metadata framework Apache Atlas. The overall proposal of goldMEDAL is synthesized in a UML class diagram.

In [42], a classification of metadata in two groups is proposed, with a special attention given to metadata related to data governance concerns such as data access, quality and security. Metadata describing various relationships among datasets are classified as inter-metadata and metadata describing datasets themselves are classified as intra-metadata. The conceptual metadata model is represented with a UML class diagram. A data lake architecture based on three zones is also represented but not formalized. This proposal has been later extended by the authors in [55] with a new analysis-oriented metadata model, also conceptualized through a UML class diagram.

Finally for metadata models, ensemble modeling and more precisely data vaults are used in [37] to create a model allowing better evolutivity for data and schema. At the conceptual level, datasets are classified inside satellites, logically abstracted inside hubs and finally associated inside links. The proposal is represented with a graph.

The storage layer of semantic data lakes has been formalized with set theory in [11] as a tuple containing a set of data sources (datasets), a set of metadata catalogs describing the datasets with directed graphs, a global knowledge graph and a mapping function relating metadata to knowledge concepts in the global graph. In the same article, the authors also propose a set-theoretical formalization of analytical queries. They are defined as sets of indicators of interest measured along sets of dimensions of analysis. A response to such query is a set of metadata and transformation rules allowing the discovery of relevant data.

In [25], the authors compare four formal schema mapping languages based on tuple-generating dependencies (tgds), namely simple tgds, nested tgds, second-order tgds (SO tgds) and plain SO tgds, as potential formal frameworks for integration tasks in data lakes. The different tgds are compared depending on their expressiveness as well as on the set of structural or reasoning properties they can ensure among the existence of universal solutions, closure under target homomorphism and allowing conjunctive query rewriting. SO tgds languages are identified as more expressive than the other two but also less reliable on the properties due to their higher time complexity for model checking.

Despite existing attempts to formalize parts of data lakes, a formal framework allowing a unified and complete representation of all the functionalities and levels of abstraction of such systems is still missing. Moreover, existing pieces of formal definitions are mainly based on semi-formal and descriptive models like UML or labeled graphs, which are not restrictive enough to guarantee mathematical, structural and/or reasoning properties to the proposed model and to ensure their preservation in any subsequent concrete implementation [6].

## 2.3 Contributions of Category Theory to Data Lakes

Category theory is an abstract meta-mathematical theory. It helps to reconcile the expressiveness of descriptive models and the restrictiveness of mathematical ones. This formalism has already been successfully used to address some challenges of computer science, for example to build a general framework for the specification of concurrent systems [14] or to allow the compositionality of machine learning components [46]. To the best of our knowledge, it has never been studied as foundation for a complete formal framework for data lakes. Nevertheless, some works tackle relevant issues to these systems with category theory.

Related to the data storage functionality, schemas and data instances in relational databases have been modeled with small categories and set-valued functors in [47], and constraints with functors and natural transformations later in [49]. Frameworks for object-oriented databases have been proposed in [30] and for document-oriented databases in [51]. The management of multi-model data and data integration issues are studied in [28, 32, 52] with some basic categorical tools like categories and functors as well as with more advanced one like pullbacks, pushforwards and kan lifts. Finally, metadata models based on category theory have been proposed in [7, 12].

On data maintenance, category theory has been mostly used to ensure data quality, through a metamodel adapted for geographic information systems in [18] and through a framework for variability models of software engineering in [35].

Finally, on data exploration, most of the existing works using category theory focus on query processing. In [22, 23], monads have been proposed as representation for queries, and monad comprehension is used for query processing. A query language implemented with the functional programming language Haskell, based on functors and using natural transformations for optimization is presented in [50]. Category theory also serves as basis for a framework of a search meta-engine in [53]. Other issues related to the data exploration fonctionality of data lakes include data and schema integration, which has been tackled in [48] with functorial data migration operations, and the creation of data visualization, which has been formalized in [54].

## 2.4 Synthesis

Data lakes have been detailed several times in terms of functionalities, but a formal definition expressed through a theoretical framework is still missing, and existing proposals either lack expressiveness or restrictiveness. Category theory is a promising candidate as foundation for such framework and has already been used in several relevant contexts for data lakes. A categorical definition unifying all the main functionalities of data lakes should provide the formal framework needed to improve them with mathematical properties and theorems, while creating a bridge with the previous works using the same formalism.

# 3 FORMALIZATION

In this section, after a brief introduction to category theory, we give a high level theoretical description of the main functionalities of a data lake presented in section 2.1. We show how composition and abstraction can be used to define different levels of representation and how different types of functors can ensure the navigation among them. We explain how our framework can be used to check the validity of an implementation of a data lake. We finally illustrate this on an example in subsection 3.3.

## 3.1 Category theory in a nutshell

Category Theory describes structures as categories and relations between them with functors.

A **category** $C$ is defined by a collection $Ob(C)$ of **objects**, a collection $Hom(C)$ of directed relations between these objects, called **morphisms**, and a binary associative operation (noted $\circ$) to compose morphisms. The sub-collection of morphisms between an object $x$ (called domain) and an object $y$ (called codomain), both in $Ob(C)$, can be expressed as $Hom_C(x, y)$, and a morphism $f$ between these objects is noted $f : x \longrightarrow y$. Each object $x \in Ob(C)$ is associated with an identity morphism $id_x : x \longrightarrow x$, acting as neutral element with $\circ$.

A category $C$ is said to be **locally small** if $Hom(C)$ is a set, **small** if $Ob(C)$ and $Hom(C)$ are both sets and **large** otherwise. There is also a large category **Cat** defined with all objects as small categories and with all morphisms as functors between them.

A **functor** $F : C \longrightarrow D$ is a structure mapping the objects and morphisms of a category $C$ to objects and morphisms of a category $D$. Functors **preserve identities**, that is $\forall x \in Ob(C), F(id_x) = id_{F(x)}$, and **preserve composition**, that is $\forall f : x \longrightarrow y, g : y \longrightarrow z, F(g \circ f) = F(g) \circ F(f)$.

A **constant functor** $\Delta_{C-D} : C \longrightarrow D$ is a special mapping that collapses every object in $Ob(C)$ to a single object $d \in Ob(D)$ and every morphism in $Hom(C)$ to the identity morphism $id_d$. **Surjective functors** act on every not empty $Hom_D(x, y)$. A functor $F : C \longrightarrow D$ is said to be surjective if for every $x, y \in Ob(D)$ and every morphism in $Hom_D(x, y)$, there is at least one morphism in $Hom_C(F^{-1}(x), F^{-1}(y))$ (surjective mapping on every morphism of D).

A **product** of two categories $C1$ and $C2$ produces a new category whose objects are all the possible pairs $(x, y)$ with $x \in \text{Ob}(C1)$ and $y \in \text{Ob}(C2)$ and morphisms $(x, y) \longrightarrow (x', y')$ are pairs $(f, g)$ where $f : x \longrightarrow x' \in \text{Hom}_{C1}(x, x')$ and $g : y \longrightarrow y' \in \text{Hom}_{C2}(y, y')$. A **bifunctor** has a product of categories as domain, and a category as codomain.

## 3.2 Categorical Framework for Data Lakes

In this subsection, we propose a high level categorical framework for data lakes, based on the description of the main functionalities established in the literature and presented in section 2.

At the highest level of abstraction, a data lake can be seen as a large category **DL**. This category is more precisely defined in table 1 and is visually represented in figure 1. Its collection of objects $Ob(\textbf{DL})$ includes three of the main functionalities identified in section 2, namely Data Storage, Data Ingestion and Data Exploration. These objects are themselves categories. The Data Maintenance functionality has a specific representation. As it mainly transforms data to improve their usability, it is represented by a bifunctor **Storage** × **Storage** ⟶ **Storage**. It allows to define maintenance operations that can have a single dataset along with its metadata as input, but also operations that take two datasets with their metadata as input (such as the discovery of relationship between entities).

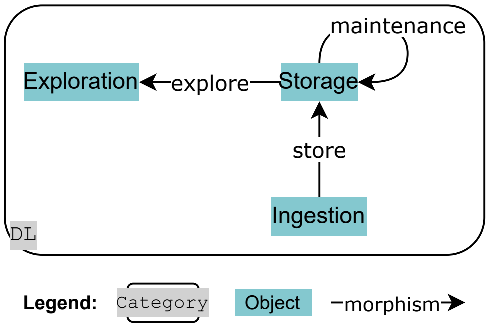


**Figure 1: Representation of the category DL**

The figure 2 and the tables 1 and 2 synthesize the following statements. The reader can use them as an helping guide of lecture.

The Data Ingestion functionality represents the entry point for data into the data lake. To do so, its **Ingestion** category contains three objects: *raw_data* for data as they are from their source, *dataset* for data as they enter into the data lake system and *metadata* that are extracted from *dataset*. The morphisms in this category show the different steps of data ingestion, i.e., we *load* the *raw_data* into the system to create a *dataset*, this *dataset* can be *transform*ed (for example to compute aggregated data or to add some lightweight information such as the timestamp of the entry in the data lake), and some *metadata* can be *extract*ed from this *dataset*.

The *dataset* and its *metadata* must be stored into the data lake. This functionality is supported by the **Storage** category. It is a category, in which objects represent data at different levels of abstraction, linked by morphisms. The different levels considered are the physical one with *md_system* and *d_system*, the logical one with *dataset*, the conceptual one with *metadata* and the structural one with *md_model* and *d_model*. The morphims link a *dataset* to its *metadata*, and the *dataset* and *metadata* to their respective *model*. The different *models* are themselves linked to their storage *system*.

| Name | Objects | Morphisms (without identity morphisms) |
|---|---|---|
| **DL** | storage, ingestion, exploration | $store : Ingestion \rightarrow Storage$<br>$explore : Storage \rightarrow Exploration$<br>$maintenance : Storage \times Storage \rightarrow Storage$ |
| **Ingestion** | raw_data, dataset, metadata | $load : raw\_data \rightarrow dataset$<br>$transform : dataset \rightarrow dataset$<br>$extract : dataset \rightarrow metadata$ |
| **Storage** | metadata, dataset, d_model, d_system md_model, md_system | $described\_by : dataset \rightarrow metadata$<br>$md\_modeled\_by : metadata \rightarrow md\_model$<br>$md\_stored\_in : md\_model \rightarrow md\_system$<br>$d\_modeled\_by : dataset \rightarrow d\_model$<br>$d\_stored\_in : d\_model \rightarrow d\_system$ |
| **Exploration** | catalogue, dataset metadata | $localize : catalogue \rightarrow dataset$<br>$query : dataset \rightarrow dataset$<br>$algorithm : dataset \rightarrow dataset$<br>$described\_by : dataset \rightarrow metadata$ |

**Table 1: High level categories of the framework**

| Name | Type | Elements in Domain | Elements in Codomain |
|---|---|---|---|
| $store$ | $Ingestion \rightarrow Storage$ | $raw\_data$<br>$dataset$<br>$metadata$<br>$load$<br>$transform$<br>$extract$ | $dataset$<br>$dataset$<br>$metadata$<br>$id_{dataset}$<br>$id_{dataset}$<br>$described\_by$ |
| $explore$ | $Storage \rightarrow Exploration$ | $dataset$<br>$d\_model$<br>$d\_system$<br>$metadata$<br>$md\_model$<br>$d\_system$<br>$described\_by$<br>$d\_modeled\_by$<br>$d\_stored\_in$<br>$md\_modeled\_by$<br>$md\_stored\_in$ | $dataset$<br>$dataset$<br>$dataset$<br>$metadata$<br>$metadata$<br>$metadata$<br>$described\_by$<br>$id_{dataset}$<br>$id_{dataset}$<br>$id_{metadata}$<br>$id_{metadata}$ |

**Table 2: Functors used in the framework**

Once the data are stored, they can be accessed in order to be explored. It can be done with the **Exploration** category. The *catalogue* allows the *localiz*ation of a *dataset*, through its *metadata*. It is possible to run some *queri*es or *algorithm*s on the retrieved *dataset* in order to get the desired result.

With this representation, constant functors can link different levels when lower level categories are embedded in objects of higher level categories. For example, the refinements carried by the **Storage** category are all embedded in one object of **DL**, namely *Storage*. A constant functor $\Delta_{Storage-DL}$ : **Storage** $\rightarrow$ **DL** can be used to link the lower level category to the higher level one. Functorial laws are satisfied because every object and every identity morphism in the domain category is mapped to the same object and its identity morphism in the codomain category (identity preservation) and because identity morphisms can always be composed with themselves (composition preservation).

To use the categories of the framework with an instance of a data lake, each implementation of a functionality must be represented in a category, that must be linked to its higher-level corresponding category with a surjective functor. The surjective condition ensures the respect of the proposed framework and its structure, while allowing more complete descriptions of functionalities in the category of the implementation. Furthermore, the morphisms of the **DL** category add a constraint that forces the existence of functors between the categories concerned by each morphism when they are represented as objects in the **DL** category.

To ensure the **navigation between the functionalities of the data lake**, functors exist between the **Ingestion** and the **Storage** categories, and between the **Storage** and the **Exploration** categories. The defined functors and morphisms force the direction of the different transformations, and avoid the scattering of data. As the categories that will be defined for the implementation must be linked to their corresponding higher-level category, this forces also the implementation to provide these functors between the different implemented functionalities.

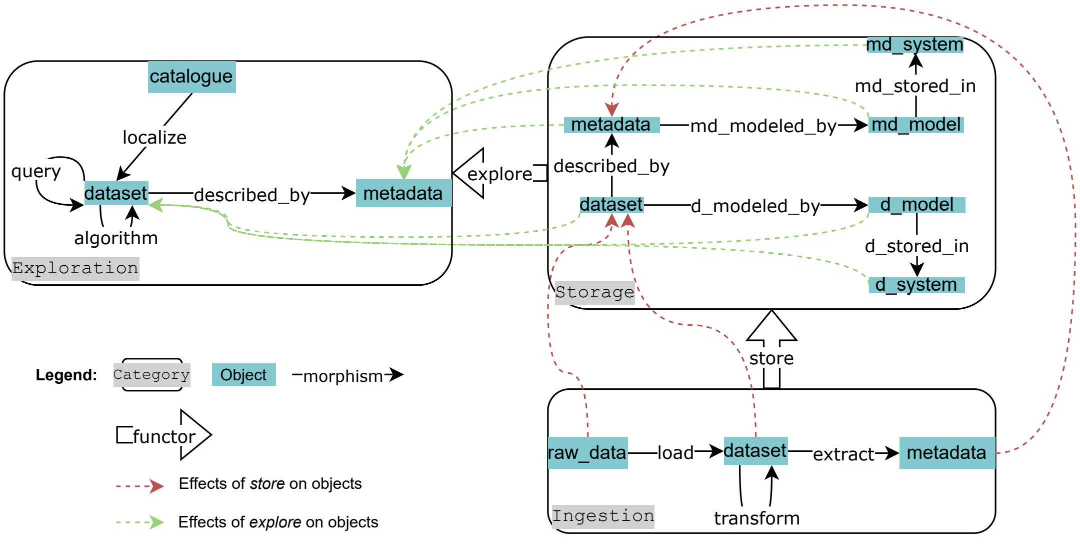


**Figure 2: Representation of the categories Ingestion, Storage and Exploration, and the functors that link them**

By relying on their constraints, such as the preservation of identities and the composition of morphisms, functors allow to **keep track of data** from their loading into the data lake to the production of results from a dataset in the exploration phase. Indeed, all the objects and the morphisms of the domain category must be sent to the codomain category. So, no information is lost when switching from a functionality to another.

The **maintenance** functionality takes a different form in the formalization framework. As it aims at improving the quality of the datasets or the metadata in order to ease their use during the exploration phase, two major applications of a maintenance operation can be found: improving a dataset or its metadata directly from themselves or improving a dataset or its metadata by relying on another dataset along with its metadata. Within the category theory, it is possible to use bifunctors to cover both of these applications. To do so, the bifunctor *maintenance* is defined as **Storage** × **Storage** → **Storage**. The first **Storage** category corresponds to the one on which the maintenance operation is applied, the second one corresponds to the one that will bring the elements required to the improvement (it can be the same **Storage** category as the first one when it is used to improve itself). The last **Storage** category is the result of the maintenance operation, and can be seen as the evolution of the first **Storage** category.

The figure 3 gives an overview of this mechanism. The objects $(dataset, metadata)$ and $(metadata, dataset)$ can be mapped either on the object $metadata$ or on the object $dataset$ of the resulting **Storage** category depending on the effect of the maintenance operation. We detail a little more this mechanism with an example in the following section. The other associations of objects are omitted for the sake of readability.

This representation **allows the composition of multiple maintenance operations**, while giving freedom to apply them in any order and to any data. Thus, the maintenance operations can be chosen according to the individual needs of each dataset. Furthermore, it has also constraints, as a **maintenance operation can only be applied on existing datasets and metadata**. This contributes to the identification of dependencies among datasets and metadata.

So, in its globality, this framework **keeps track of the entire lineage of data**. It is possible to know the source of data, how they have been transformed and the relationships that exist among datasets. It also imposes **constraints** on the validity of transformations applied on data, and on the switching from a functionality to another. Furthermore, **operations can be composed**, mainly with the *maintenance* functor, in order to allow a high flexibility that is essential to data lakes.

## 3.3 Example

We propose to show the usability of our categorical framework by defining a small data lake. We rely on the following use case: an enterprise has data about their customers, and records their online activity on the enterprise web applications.

The instances of the **Ingestion** and **Storage** categories are represented in figure 4. Regarding the data of the online activity, the enterprise is not interested by all the data, so it performs an aggregation operation in order to store only a summary of the data (category **Ing_ds1**). The data of the customers are easier to handle. As they are extracted from the enterprise information system, they do not need any transformation before being stored into the data lake (category **Ing_ds2**).

Once the data are ingested, they are stored in the data lake. For the activity data, the dataset is modeled as time series and InfluxDB is used as storage system. For the metadata, they are modeled as graphs in Neo4j (category **Str_ds1**). Regarding the customer data, the metadata are stored in the file system as JSON format, and the dataset in the relational PostgreSQL database (category **Str_ds2**).

The table 3 states the effects of functors on the objects and morphisms from the instance categories of the figure 4 to the corresponding high-level categories **Ingestion** and **Storage**. Thus, the ingestion and storage functionalities of the implemented data lake

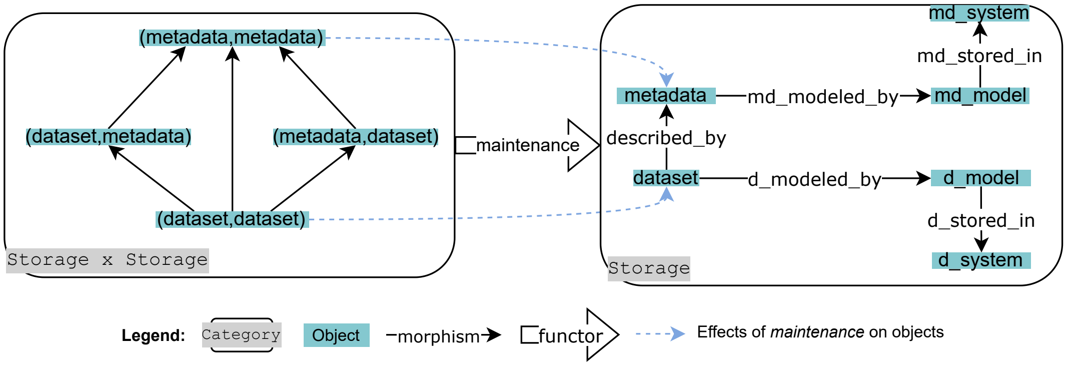


**Figure 3: Partial representation of the *maintenance* functor, in which the immutable parts are represented**

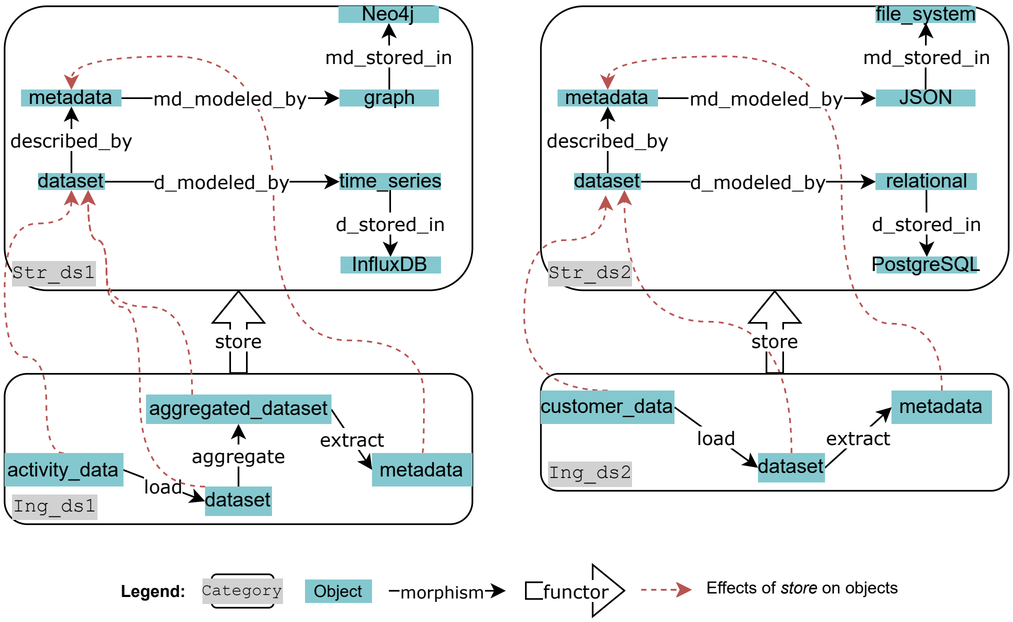


**Figure 4: The ingestion and storage of the two datasets**

satisfy the requirements of the formalization including the surjectivity condition on functors.

Once the two datasets are stored, a maintenance operation can be applied on them (figure 5). In this operation, the dataset of the temporal activity is enriched with the data about customers (category **Str_ds1_v2**) in order to gain more information regarding their different characteristics, for example their country that can be used to make the typical hours of activity more precise. With this enriched dataset, an exploration is performed, first to reduce the dataset on a given period, and then to execute an anomaly detection algorithm to reveal fraudulent uses.

This example demonstrates how the category theory supports the navigation across abstraction levels and how properties on functors constrain the implemented functionalities to comply to the structure defined in the corresponding higher-level categories. Moreover, the framework allows to represent all the potential functionalities of a data lake, and thus to use it to check the validity of an implementation of a data lake.

## 4 CONCLUSION

In this article, we have proposed a unified formal framework for data lakes based on category theory. The navigation between the different functionalities is controlled by functors and compositions, that allow to keep track of the lineage of data while providing the flexibility required by these systems. The levels of abstraction of the data lake are linked with constant or surjective functors, that ensure the validity of implementations of data lakes. We have shown on an example how the framework can be used.

Unlike previous works on the formalization of data lakes, our proposal considers a unified and complete view of all the main functionalities identified by the literature, namely Data Ingestion,

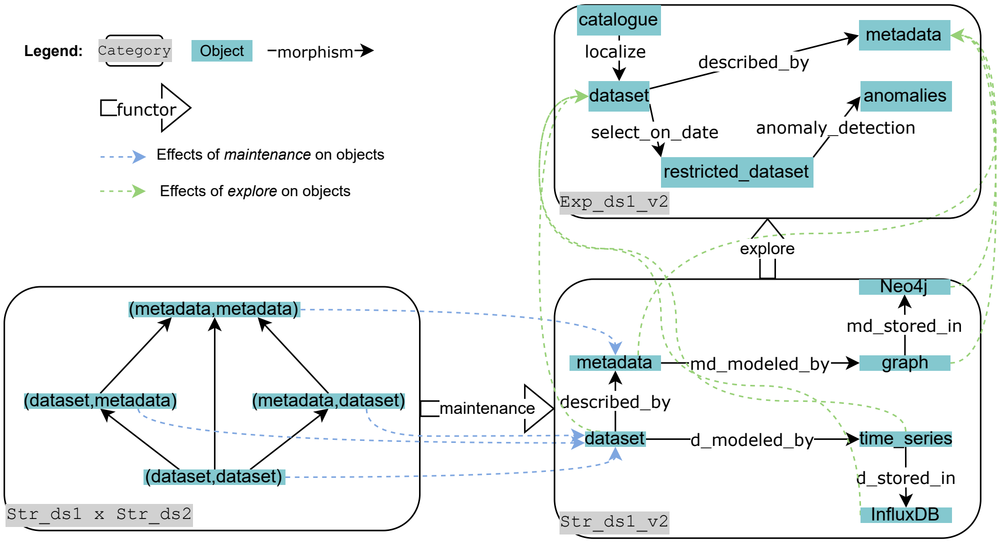


**Figure 5: A maintenance operation followed by an exploration**

Data Storage, Data Maintenance and Data Exploration. Thanks to the expressiveness and restrictiveness of category theory, we have also been able to represent and control the various dependencies and levels of abstraction existing in data lakes. Category theory additionally creates a bridge with existing works of the literature using the same formalism, allowing their use as refinements of some higher level aspects described in our framework.

As perspectives for future works, we plan to extend our formalism to allow the definition and control of complex and hybrid workflows for accessing and querying data in data lakes. Such workflows are indeed very important for these systems, in which a variety of operations can be executed in the same environment like for example operators of relational algebra, machine learning tasks based on linear algebra, user-defined functions, etc. These various operations could therefore be unified by expressing them through categories, functors and bifunctors and then linked to the rest of the framework. We also plan to introduce the physical level of components in the data lake architecture by mapping the implemented functionalities to their corresponding component through functors. With this configuration, we can rely on a previous work [21] that allows to check the conservation or the loss of technical properties in an architecture with the category theory. We also think about exploring more the capabilities of the *maintenance* functor, that could be used to control the models of the dataset and the metadata depending on their original models with the $(d\_model, d\_model)$ object of the product of categories. It can serve to detect model transformations that will lose precision compared to the original model.

## REFERENCES

[1] Ayman Alserafi, Alberto Abelló, Oscar Romero, and Toon Calders. 2016. Towards information profiling: data lake content metadata management. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 178–185.

[2] Michael Armbrust, Tathagata Das, Liwen Sun, Burak Yavuz, Shixiong Zhu, Mukul Murthy, Joseph Torres, Herman van Hovell, Adrian Ionescu, Alicja Łuszczak, et al. 2020. Delta lake: high-performance ACID table storage over cloud object stores. *Proceedings of the VLDB Endowment* 13, 12 (2020), 3411–3424.

[3] Michael Armbrust, Ali Ghodsi, Reynold Xin, and Matei Zaharia. 2021. Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In *Proceedings of CIDR*.

[4] Amin Beheshti, Boualem Benatallah, Reza Nouri, and Alireza Tabebordbar. 2018. CoreKG: a knowledge lake service. *Proceedings of the VLDB Endowment* 11, 12 (2018), 1942–1945.

[5] Manfred Broy. 2011. Can practitioners neglect theory and theoreticians neglect practice? *Computer* 44, 10 (2011), 19–24.

[6] Manfred Broy and María Victoria Cengarle. 2011. UML formal semantics: lessons learned. *Software & Systems Modeling* 10, 4 (2011), 441–446.

[7] Isabel Cafezeiro and Edward Hermann Haeusler. 2007. Semantic Interoperability via Category Theory.. In *ER (Tutorials, Posters, Panels & Industrial Contributions)*. Citeseer, 197–202.

[8] Edgar Frank Codd. 1983. A relational model of data for large shared data banks. *Commun. ACM* 26, 1 (1983), 64–69.

[9] Julia Couto, Olimar Teixeira Borges, Duncan D Ruiz, Sabrina Marczak, and Rafael Prikladnicki. 2019. A Mapping Study about Data Lakes: An Improved Definition and Possible Architectures.. In *SEKE*. 453–578.

[10] Zhamak Dehghani. 2019. How to move beyond a monolithic data lake to a distributed data mesh. *Martin Fowler's Blog* (2019).

[11] Claudia Diamantini, Domenico Potena, and Emanuele Storti. 2021. A Semantic Data Lake Model for Analytic Query-Driven Discovery. In *The 23rd International Conference on Information Integration and Web Intelligence*. 183–186.

[12] Zinovy Diskin. 1997. The Arrow Logic of Metadata Environment: A Formalised Graph-Based Framework for Structuring Metadata Repositories. (1997).

[13] James Dixon. 2010. Pentaho, Hadoop, and data lakes. *blog, Oct* (2010).

[14] Hartmut Ehrig, Martin Große-Rhode, and Uwe Wolter. 1998. Applications of category theory to the area of algebraic specification in computer science. *Applied categorical structures* 6, 1 (1998), 1–35.

[15] Samuel Eilenberg and Saunders MacLane. 1945. General theory of natural equivalences. *Trans. Amer. Math. Soc.* 58, 2 (1945), 231–294.

[16] Mina Farid, Alexandra Roatis, Ihab F Ilyas, Hella-Franziska Hoffmann, and Xu Chu. 2016. CLAMS: bringing quality to data lakes. In *Proceedings of the 2016 International Conference on Management of Data*. 2089–2092.

[17] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1001–1012.

[18] Andrew U Frank. 1998. Metamodels for data quality description. *Data Quality in Geographic Information-From Error to Uncertainty* 192 (1998).

[19] Yihan Gao, Silu Huang, and Aditya Parameswaran. 2018. Navigating the data lake with datamaran: Automatically extracting structure from log datasets. In *Proceedings of the 2018 International Conference on Management of Data*. 943–958.

[20] I. Gartner. 2014. Gartner Says Beware of the Data Lake Fallacy. https://www.gartner.com/newsroom/id/2809117.

[21] Annabelle Gillet, Éric Leclercq, and Nadine Cullot. 2021. Lambda+, the Renewal of the Lambda Architecture: Category Theory to the Rescue. In *International*

| Functor | Element in domain | Element in codomain |
|---|---|---|
| $Ing\_ds1 \rightarrow Ingestion$ | $activity\_data$ | $raw\_data$ |
| | $dataset$ | $dataset$ |
| | $aggregated\_dataset$ | $dataset$ |
| | $metadata$ | $metadata$ |
| | $load$ | $load$ |
| | $aggregate$ | $transform$ |
| | $extract$ | $extract$ |
| $Ing\_ds2 \rightarrow Ingestion$ | $customer\_data$ | $raw\_data$ |
| | $dataset$ | $dataset$ |
| | $metadata$ | $metadata$ |
| | $load$ | $load$ |
| | $id_{dataset}$ | $transform$ |
| | $extract$ | $extract$ |
| $Str\_ds1 \rightarrow Storage$ | $dataset$ | $dataset$ |
| | $time\_series$ | $d\_model$ |
| | $InfluxDB$ | $d\_system$ |
| | $metadata$ | $metadata$ |
| | $graph$ | $md\_model$ |
| | $Neo4j$ | $d\_system$ |
| | $described\_by$ | $described\_by$ |
| | $d\_modeled\_by$ | $d\_modeled\_by$ |
| | $d\_stored\_in$ | $d\_stored\_in$ |
| | $md\_modeled\_by$ | $md\_modeled\_by$ |
| | $md\_stored\_in$ | $md\_stored\_in$ |
| $Str\_ds2 \rightarrow Storage$ | $dataset$ | $dataset$ |
| | $relational$ | $d\_model$ |
| | $PostgreSQL$ | $d\_system$ |
| | $metadata$ | $metadata$ |
| | $JSON$ | $md\_model$ |
| | $file\_system$ | $d\_system$ |
| | $described\_by$ | $described\_by$ |
| | $d\_modeled\_by$ | $d\_modeled\_by$ |
| | $d\_stored\_in$ | $d\_stored\_in$ |
| | $md\_modeled\_by$ | $md\_modeled\_by$ |
| | $md\_stored\_in$ | $md\_stored\_in$ |

**Table 3: Functors used in the example, between the category representing the instance of a functionality and the high-level category of the functionality**

*Conference on Advanced Information Systems Engineering.* Springer, 381–396.

[22] Georg Gottlob and Christoph Koch. 2002. Monadic queries over tree-structured data. In *Proceedings 17th annual IEEE symposium on logic in computer science.* IEEE, 189–202.

[23] Torsten Grust. 2004. Monad comprehensions: a versatile representation for queries. In *The Functional Approach to Data Management.* Springer, 288–311.

[24] Rihan Hai, Sandra Geisler, and Christoph Quix. 2016. Constance: An intelligent data lake system. In *Proceedings of the 2016 international conference on management of data.* 2097–2100.

[25] Rihan Hai, Christoph Quix, and Matthias Jarke. 2021. Data lake concept and systems: a survey. *arXiv preprint arXiv:2106.09592* (2021).

[26] Rihan Hai, Christoph Quix, and Dan Wang. 2019. Relaxed functional dependency discovery in heterogeneous data lakes. In *International Conference on Conceptual Modeling.* Springer, 225–239.

[27] Alon Y Halevy, Flip Korn, Natalya Fridman Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. 2016. Managing Google's data lake: an overview of the Goods system. *IEEE Data Eng. Bull.* 39, 3 (2016), 5–14.

[28] Irena Holubova, Pavel Contos, and Martin Svoboda. 2021. Categorical Management of Multi-Model Data. In *25th International Database Engineering & Applications Symposium.* 134–140.

[29] Pontus Johnson, Mathias Ekstedt, and Ivar Jacobson. 2012. Where's the theory for software engineering? *IEEE software* 29, 5 (2012), 96–96.

[30] P Kolencık. 1998. *Categorical Framework for Object-Oriented Database Model.* Ph. D. Dissertation. PhD thesis.

[31] Pengfei Liu, Sabine Loudcher, Jérôme Darmont, and Camille Noûs. 2021. ArchaeoDAL: A Data Lake for Archaeological Data Management and Analytics. In *25th International Database Engineering & Applications Symposium.* 252–262.

[32] Zhen Hua Liu, Jiaheng Lu, Dieter Gawlick, Heli Helskyaho, Gregory Pogossiants, and Zhe Wu. 2018. Multi-model database management systems-a look forward. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare.* Springer, 16–29.

[33] Antonio Maccioni and Riccardo Torlone. 2018. KAYAK: a framework for just-in-time data preparation in a data lake. In *International Conference on Advanced Information Systems Engineering.* Springer, 474–489.

[34] Jacob McPadden, Thomas JS Durant, Dustin R Bunch, Andreas Coppi, Nathan Price, Kris Rodgerson, Charles J Torre Jr, William Byron, H Patrick Young, Allen L Hsiao, et al. 2018. A scalable data science platform for healthcare and precision medicine research. *arXiv preprint arXiv:1808.04849* (2018).

[35] Daniel-Jesus Munoz, Dilian Gurov, Monica Pinto, and Lidia Fuentes. 2021. Category Theory Framework for Variability Models with Non-functional Requirements. In *International Conference on Advanced Information Systems Engineering.* Springer, 397–413.

[36] Amr A Munshi and Yasser Abdel-Rady I Mohamed. 2018. Data lake lambda architecture for smart grids big data analytics. *IEEE Access* 6 (2018), 40463–40471.

[37] Iuri D Nogueira, Maram Romdhane, and Jérôme Darmont. 2018. Modeling data lake metadata with a data vault. In *Proceedings of the 22nd International Database Engineering & Applications Symposium.* 253–261.

[38] Christoph Quix, Rihan Hai, and Ivan Vatov. 2016. Metadata extraction and management in data lakes with GEMMS. *Complex Systems Informatics and Modeling Quarterly* 9 (2016), 67–83.

[39] Raghu Ramakrishnan, Baskar Sridharan, John R Douceur, Pavan Kasturi, Balaji Krishnamachari-Sampath, Karthick Krishnamoorthy, Peng Li, Mitica Manu, Spiro Michaylov, Rogério Ramos, et al. 2017. Azure data lake store: a hyperscale distributed file service for big data analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data.* 51–63.

[40] Sarathkumar Rangarajan, Huai Liu, Hua Wang, and Chuan-Long Wang. 2015. Scalable architecture for personalized healthcare service recommendation using big data lake. In *Service research and innovation.* Springer, 65–79.

[41] Franck Ravat and Yan Zhao. 2019. Data lakes: Trends and perspectives. In *International Conference on Database and Expert Systems Applications.* Springer, 304–313.

[42] Franck Ravat and Yan Zhao. 2019. Metadata management for data lakes. In *European Conference on Advances in Databases and Information Systems.* Springer, 37–44.

[43] David Sarramia, Alexandre Claude, Francis Ogereau, Jérémy Mezhoud, and Gilles Mailhot. 2022. CEBA: A Data Lake for Data Sharing and Environmental Monitoring. *Sensors* 22, 7 (2022), 2733.

[44] Pegdwendé Sawadogo and Jérôme Darmont. 2021. On data lake architectures and metadata management. *Journal of Intelligent Information Systems* 56, 1 (2021), 97–120.

[45] Etienne Scholly, Pegdwendé Sawadogo, Pengfei Liu, Javier Alfonso Espinosa-Oviedo, Cécile Favre, Sabine Loudcher, Jérôme Darmont, and Camille Noûs. 2021. Coining goldMEDAL: a new contribution to data lake generic metadata modeling. *arXiv preprint arXiv:2103.13155* (2021).

[46] Dan Shiebler, Bruno Gavranović, and Paul Wilson. 2021. Category theory in machine learning. *arXiv preprint arXiv:2106.07032* (2021).

[47] David Spivak. 2011. *Categorical Information Theory.* Technical Report. Massachusetts Inst. of Tech.

[48] David I Spivak. 2012. Functorial data migration. *Information and Computation* 217 (2012), 31–51.

[49] David I Spivak. 2014. Database queries and constraints via lifting problems. *Mathematical structures in computer science* 24, 6 (2014).

[50] Laurent Thiry, Heng Zhao, and Michel Hassenforder. 2018. Categories for (Big) Data models and optimization. *Journal of Big Data* 5, 1 (2018), 1–20.

[51] David Toth. 2008. Database engineering from the category theory viewpoint. *Databases, Texts* (2008), 37.

[52] Valter Uotila and Jiaheng Lu. 2021. A Formal Category Theoretical Framework for Multi-model Data Transformations. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare.* Springer, 14–28.

[53] Paul-Christophe Varoutas, Philippe Rizand, and Alain Livartowski. 2006. Using category theory as a basis for a heterogeneous data source search meta-engine: the Prométhée framework. In *International Conference on Algebraic Methodology and Software Technology.* Springer, 381–387.

[54] Paul Vickers, Joe Faith, and Nick Rossiter. 2012. Understanding visualization: A formal approach using category theory and semiotics. *IEEE transactions on visualization and computer graphics* 19, 6 (2012), 1048–1061.

[55] Yan Zhao, Imen Megdiche, Franck Ravat, and Vincent-nam Dang. 2021. A Zone-Based Data Lake Architecture for IoT, Small and Big Data. In *25th International Database Engineering & Applications Symposium.* 94–102.

# On Extended Nested Relational Schemas Generated by Context-free Grammars

András Benczúr
Eötvös Loránd University, Budapest, Hungary
abenczur@inf.elte.hu

Gyula I. Szabó
Eötvös Loránd University, Budapest, Hungary
gyula@szaboo.de

## ABSTRACT

The novelty of the paper is that we introduce extended relational schemas defined by context-free languages. This allows us to specify schemas that include nested relations. The representation of regular expressions with the graph made it possible to specify both relational algebraic operations and dependencies. This paper uses extended context-free languages to define schema graphs, constructed from the regular expressions specifying the right side of the production rules. This is exploited by representing context-free grammars with recursive, increasing graphs. Next, we introduce the use of a context-free language to specify extended relational schemas with ordinary and nested attributes. In the derivation rules of the grammar, terminal symbols give ordinary attributes, while non-terminal (linguistic) symbols give nested attributes. Regular expressions can use both terminal and linguistic symbols. We can define schemas with finite derivations of rules. In the accepted symbol sequences, the terminal symbols give the ordinary attributes and the language symbols the nested attributes. For a nested attribute we must assign a schema with the appropriate rule. No embedded attribute may remain at the end. The relational row type and the finite occurrence of rows of this type can be defined as instance for the resulting schema. We define set operations and nested database operations as well. Functional dependencies can be defined and examined at multiple depths.

## CCS CONCEPTS

• **Information systems** → Data management systems; Database design and models; Graph-based database models.

## KEYWORDS

Regular Language, Context-free Language, XML, Data Models, Database Design

**ACM Reference Format:**
András Benczúr and Gyula I. Szabó. 2022. On Extended Nested Relational Schemas Generated by Context-free Grammars. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3548785.3548795

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548795

## 1 INTRODUCTION

XML has been practically the standard format of data exchange over the World Wide Web. In XML, you can specify a structure type by a formal language. Instances of a particular type of XML Element are can be considered a data collection. The declaration of a DTD Element consisting of simple values can be considered as a general relational schema definition. An instance from the extended schema is a row type from the language given by the regular expression. Occurrences of an Element corresponding to a row type of a relation make a relation instance. The declaration of a DTD Element consisting of simple values can be considered as a general relational schema definition. Such an extension is included in [2], [3].

This can be extended to complex Element declarations given by a context-free grammar. For this, we use two types of Element names in the DTD. Simple type, like in the regular case, can only take simple values. The other is the complex type, for which the regular expression of the declaration specifies a choice of row type, where simple and complex element names can be included in the allowed row type. The set of Element declarations in the DTD thus represents a context-free language, where simple names are the terminal symbols and complex names are the non-terminals. For each complex element we can associate a language built in parallel. A sentence from these languages consists of a list of simple element names and specifies a row type.

The use of the resulting regular schemas is illustrated by the graph of the finite-state automation that can be given directly from a regular grammar, or as it is given in [3], by graphs that can be directly assigned to regular expressions. This formal language method, which specifies the schema system, can be applied to context-free languages. The first option seemed to be to determine context-free languages with graphs of recursive finite state automata. This is a less usable representation than the direct use of graphs of regular expressions in the version of context-free grammars given by regular expressions. We build on this idea when introducing nested relational extended schemas based on the use of context-free languages.

XML was originally defined for describing and presenting individual documents, but it has been used for building databases too. Because of the use of XML as database model, one needs XML integrity constraints, and XML functional dependency concepts. The main problem with defining functional dependency in the XML context is the lacking "tuple" concept for XML. An instance of a relational schema is a set of tuples, and one can easily select pairs of tuples from this set for comparing in order to check whether the instance satisfies a given functional dependency defined on the relational schema. In the XML world, there is no general accepted definition for the concept of tuple, and even if one chooses a collection of elements and declares them a "tuple", it is very hard to find a proper matching algorithm for them. Arenas and Libkin defined "tree tuples" in their seminal work [4], based upon DTD schema. Vincent et al. [5] described some cases, not covered with "tree tuples", and invented the notion "closest node" to deal with them. They defined functional dependency on XML trees without any schema, and used DTDs just to prove that their definition is equivalent with "tree tuples" for some classes of DTDs.

All XFD concepts are very intricate, compared with the classical functional dependency concept for relational databases. In the case of XML data model, they base mostly upon path expressions.

A new functional dependency concept, regular FD has been proposed recently [2], applicable for data models, those extended "tuples" are sentences from a regular language. The main motivation was to find a simple, but general definition of functional dependency for a broad family of data models: the only assumption was that the "tuples" should be sentences of a given regular language (i.e., they should be generated by a regular grammar).

In this paper, we rephrase the concept of regular relational schema to the schema graph that can be constructed for extended context-free grammars. The schema graph can be used to expound instances of relational databases with complex values and some algebraic operations as well. Using the schema graph we can define scoped functional dependency on extended context-free languages.

## 2 GRAPH REPRESENTATION FOR FORMAL LANGUAGES

We want to generalize the definition of relation for a broad family of data models: the only assumption is that the "tuples" should be sentences of a given formal language (i.e., they should be generated by a grammar). Our more general new model makes use of formal languages to specify „tuple" types, these schemas will be given as sentences of a formal language. In order to realize this aim, it is helpful to create graph representation for formal languages. A regular language can be represented by the graph of the corresponding ((non)deterministic) finite state automaton (FSA). Alg. 1. in [2] can create this automaton from the grammar of the language, If the regular language is given by a regular expression, then there exist a great number of algorithms for the efficient construction of a finite automaton from a given regular expression. There are two main types of them according to the working-method of the resulting state machines: non-deterministic (NFA, e.g. Glushkov automaton) and deterministic (DFA, e.g. Brzozowski's construction). The classical algorithm of Berry and Sethi [6] constructs efficiently a DFA from a regular expression when all symbols are distinct. We use here another algorithm to construct a graph representation from a regular expression ( [3])

A graph representation of a regular language is an edge-, or node-labeled directed graph with one entry point and one exit point. Routes from the point of entry to the point of exit are called traversals. The series of labels in the traversals make up the language.

For regular languages, we can obtain a graph representation in two ways. For a language given by a regular grammar we can directly construct the corresponding finite state automaton. In the other case where the language is given by a regular expression we can construct an edge-labeled graph directly from regular expressions. One such construct is the [6] graph, the other is the graph in the [1].

### 2.1 Graph Representation for Regular Expressions

**Definition 1.** (Regular Expression Syntax). Let X be a finite set of symbols (alphabet), then a regular expression RE over X (denoted by $RE_X$, or simply RE, if X is understood from the context) is recursively defined as follows:

RE::=0 | 1 | $\alpha$ | RE + RE | RE $\circ$ RE | RE* | RE$^?$ , where $\alpha$ is in X.

The regular expression RE generates the regular language L(RE). L(0) is the empty language, L(1) is the language consisting of the empty string $\epsilon$ alone. Note that 0 and 1 are not symbols from the alphabet X: 0 represents the empty regular expression, 1 represents the empty string $\epsilon$

We need a construction for the graph representation of regular expressions. We will construct a graph from vertices picked from a suitably large symbol set Γ. We assume that {IN,OUT} ⊆ Γ and by picking a node v ∈ Γ we remove it from Γ. The vertices IN and OUT get the labels IN and OUT, respectively.

**Algorithm 1** Construction of the Graph-Representation for a regular expression according to Def. 1.

Input: regular expression RE (built from the alphabet Σ),
Output: vertex labeled digraph G(RE)=(V,E) representing RE.
if RE=0, then V= ∅ and E= ∅.
if RE=1, then V={IN,OUT} and E={(IN,OUT)}.
if RE=A,A ∈ Σ, then we pick a node v ∈ Γ, set V={IN,OUT,v}, and E={(IN,v), (v,OUT)}. We label the node v with A.
if $RE_1$ and $RE_2$ are regular expressions, then G($RE_1$+$RE_2$) will be formed by uniting the IN and OUT nodes of G($RE_1$) and G($RE_2$), respectively.
if $RE_1$ and $RE_2$ are regular expressions, then in order to build the graph G($RE_1$ $\circ$ $RE_2$) we first rename the OUT node of G($RE_1$) and the IN node of G($RE_2$) to JOIN, then unite them using the JOIN node as a connecting switch in order to get a more compact graph (Fig. 1).
if RE is a regular expression and G(RE)=(V,E), then G(RE$^?$)=(V,E ∪ (IN,OUT)).
if RE is a regular expression, then in order to build the graph G(RE*) we first pick a node v ∈ Γ, then we create the graph G*(RE)=G(RE) ∪ {v} (It means that V*=V ∪ {v}, the node v gets the special label STAR). Let us denote {$a_1$,...,$a_n$} the nodes with ingoing edge from IN and {$z_1$,...,$z_n$} the nodes with outgoing edge to OUT, respectively. Let us create the graph $G_{IN}$ (RE,STAR)=$\cup_1^n(v, a_i)$ and the graph $G_{OUT}$ (RE,STAR)=$\cup_1^n(z_i, v)$, respectively. Then G(RE*)=G*(RE) ∩ $G_{IN}$(RE,STAR) ∪ $G_{OUT}$ (RE,STAR) ∪ (IN,STAR) ∪ (STAR,OUT.

Theorem 1. ( [1], [3])

The (IN,....,OUT) traversals on the graph representation G(RE) for the regular expression RE constructed by Alg. 1. generate exactly the regular language L(RE) over the alphabet Σ of RE.

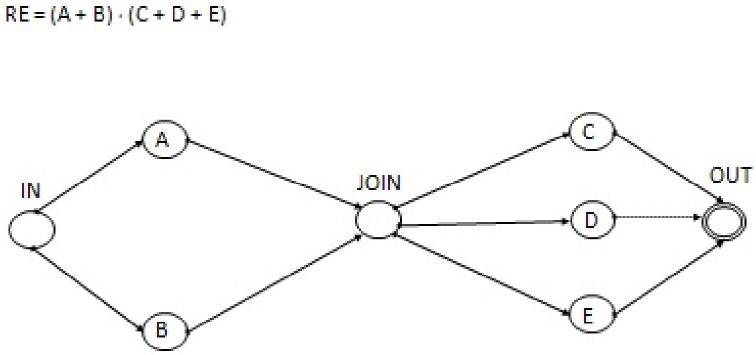


**Figure 1: Using the JOIN node as connecting switch for the concatenation of two RE graphs**

## 2.2 Graph Representation for Context-free Languages by RSM Extension

In the case of context-free languages, specifying a graph representation is a complex task. It is not sufficient to specify a single graph but we need to give possibly several graphs that call each other recursively. Such a representation can be find in [7], the graphs of the so-called recursive finite state machine. An interesting use of RSM representations can be found in [8]. The authors use graph operations and matrix algorithms in evaluation of context-free path queries.

Next, we introduce recursive state machines (RSM) from [8].

This kind of computational machine extends the definition of finite state machines and increases the computational capabilities of this formalism.

A recursive state machine R over a finite alphabet Σ is defined as a tuple of elements $(M, m, \{C_i\}_{i \in M})$ where:

-{M} is a finite set of labels of boxes.

-m ∈ M is an initial box label.

-Set of component state machines or boxes, where $C_i = (\Sigma \cup M, Q_i, q_i^0, F_i, \delta_i)$ :

-$\Sigma \cup M$ is a set of symbols, $\Sigma \cap M = \emptyset$

-$Q_i$ is a finite set of states, where $Q_i \cap Q_j = \emptyset, \ \forall i \neq j$

- $q_i^0$ is an initial state for the component state machine $C_i$

-$F_i$ is a set of final states for $C_i$.where $F_i \subseteq Q_i$

-$\delta_i$ is a transition function for $C_i$, where $\delta_i : Q_i \times (\Sigma \cup M) \rightarrow Q_i$

RSM behaves as a set of finite state machines (or FSM). Each FSM is called a box or a component state machine [7]. A box works almost the same as a classical FSM, but it also handles additional recursive calls and employs an implicit call stack to call one component from another and then return execution flow back. RSMs are equivalent to context-free languages.

A version of the RSM construction will be given in the following. To do this, we start by introducing a special form of context-free grammars. Each language symbol is assigned a single regular grammar whose terminal symbols are the terminal and linguistic symbols of the original grammar.

Definition 2.

G = (N, T, S, P) is a context-free grammar given by regular languages, where T is the set of terminal, N is the finite set of non-terminal symbols, P is the set of production/derivation rules, and S is the sentence symbol. The right side of production rules are given by regular languages: for all p in P, p = {A => w | w in $L(G_A)$, A in N, $G_A = \{N \cup T, N_A, P_A, S_A\}$ is a regular grammar}.

During the construction of graph representations for each language symbol, we shall build the finite state automata with expansion iteration. We will get graphs that have edges labeled by element of T and N. The restriction of an F graph on the terminals is denoted by $^{T}F$.

First, for each A ∈ N, we construct a finite-state automaton for its rules, $M_A$, with $Start_A$ and $End_A$ states of entering and leaving respectively.

Let’s start with the graph $M_S$.

The first sub-language, $L(^{T}M_S)$, is obtained by erasing all edges with non-terminal labels and taking the language generated by the remaining automaton. If this language is not empty, then each sentence can be directly produced from S, so it is an element of the language L(G).

Iteration step: We have constructed the $F_i$ graph and the corresponding $L(^{T}F_i)$ language. Choose an edge (p, q) in $F_i$ with an A∈N label. Insert the graph $p \rightarrow M_A \rightarrow q$ parallel with the A edge, where $M_A$ is the graph of the last extended automaton of A. Do this for all occurrences of edges with label A. This will be the $F_{i+1}$ graph. The new language will be $L(^{T}F_{i+1})$. Obviously $L(^{T}F_{i+1})$ contains $L(^{T}F_i)$. The vertex labels of graphs are defined by the non-terminal symbols of regular grammars in the rules. Repeated use of vertex labels is not a problem with pastes. For example, the language L(G) can be produced as an expanding series of regular languages generated by extending the graph $M_S$. The procedure is not deterministic, but it can be made deterministic by always inserting new $M_A$ graphs in a cyclic order of N.

For each regular grammar $G_A$, we can build the extensions of the RSM graph to produce the corresponding language over terminals N. The notation of these languages is $L(M_A)$.

We can generate regular expressions equivalent to $G_A$ grammar and we denote them by $E_A(N∪T)$. By writing the language $L(M_A)$ in the place of A in a regular expression, we get regular expression above the languages. Formally, by using the mapping $f(A)=L(M_A)$, we get the expression $E(f(N) \cup T)$ over the languages $\{L(M_A)| A∈N\}$ from the expression $E(N∪T)$. The notation f(N) is interpreted as meaning that any element of the language $L(M_A)$ can be substituted in place of the occurrence of A in the regular expression.

**Theorem 2.** The languages $L(G)= L(M_S)$ and $\{L(M_A)| A∈N\}$ satisfy the following system of equations:

$L(M_A)= E_A(f(N) \cup T)$, for all A∈N.

Proof.

Let we take a specific rule A => $w∈L(G_A)$, where w = vBz, and suppose that exists x $∈L(M_B)$ such that vxz $\in L(M_A)$. There is a traversal of the graph $M_A$ according to w, including the B edge with non-terminal label. Any graph can be written in place of the edge B, which can be obtained during the extension of the graph $M_B$. This means that any sentence of the language $L(M_B)$ can be inserted here as a path, i.e. the whole language $L(M_B)$ can be inserted in place of B.

## 2.3 Graph Representation for Extended Context-free Languages

There are other ways to get to infinite graphs by specifying the set of rules for individual non-terminals by regular expressions instead of regular grammars. In Alg. 2. we shall use the graph constructed by Alg. 1. for the regular expression. You can insert the appropriate graph in place of the non-terminal vertices, so you can get newer, longer traversals. All the traversals through the infinite graph obtained by infinitely extending the graph of the sentence symbol again give the sentences of the language.

An extended context-free language (ECFL) is generated by an extended context-free grammar (ECFG). An ECFG is a tuple G=(N,T,R,P), where

N is the (finite) set of non-terminal symbols,

T is the (finite) set of terminal symbols, and $N \cap T = \phi$,

P is the set of production rules of the form A => $R_A$, where A ∈ N, $R_A$ is a regular expression over $N \cup T$,

R ∈ N is the start symbol.

For each production rule A => $R_A$ the regular expression $R_A$ denotes a regular language $L_A \subseteq (N \cup T)^*$, the corresponding graph-representation can be constructed according to Alg. 1., using as input alphabet $\Sigma = N \cup T$. During derivation by the grammar G we substitute the non-terminal A by a sentence from $L_A$, so the vertex-labels set by Alg. 1. will be picked from either N or T. In order to demonstrate the path that leads to the attributes of a schema we use the list of non-terminal symbols that will be used during the derivation process leading to a terminal symbol. In the first step we use the start symbol R as the beginning. In order to create a graph for the extended context-free language L(G) generated by the grammar G we should repeat the construction for all vertices, labeled with a symbol X ∈ N.

**Algorithm 2** Construction of the infinite graph for an extended context-free language.

Input: an ECFG G=(N,T,R,P),

Output: vertex labeled digraph SCH(G)=(V,E) representing the graph for G.

Step 1. We apply Alg. 1. for the RE $R_R$, using $N \cup T$ as alphabet $\Sigma$ and vertex-label R.Y in state of symbol Y ($Y \in N \cup T$). We denote the created graph as $G_1$. If all vertex-labels in $G_1$ have a terminal symbol as the last component, then the construction is ready and SCH(G)= $G_1$. If there are vertices in $G_1$ with non-terminal symbol as the last component of their label, then these vertices will be bracketed: in the ingoing edge of the vertex we insert a node with label ***in***, and in the outgoing edge a node with label ***out***, respectively. These ***in*** and ***out*** labels do not have dotted form. (see Fig. 2)

Step k. (k ≥ 2). If all vertex-labels in $G_{k-1}$ have a terminal symbol as the last component (ordinary attributes), then the construction is ready and SCH(G)= $G_{k-1}$. Otherwise, we pick a vertex whose label ends to a non-terminal symbol, say Z. This vertex has been (***in***, ***out***) bracketed during the construction Step k-1. We apply Alg. 1. for the RE $R_Z$, using vertex-label R...Z.Y in state of symbol Y ($Y \in N \cup T$). We select from the created graph a sub graph that consists of complete (IN,. . .,OUT) paths. Let this sub graph be $G^Z$. The vertices in $G^Z$ with non-terminal symbol as the last component of their label will be bracketed similarly as described in Step 1. (see Fig. 3. and Fig. 5). Let the vertex v in $G_{k-1}$ be the picked one, labeled with the non-terminal Z. Let $(v_{in},v)$ and $(v, v_{out})$ be the ingoing and outgoing edges to v ($v_{in}$ has the label ***in***, $v_{out}$ has the label ***out***), then we unite the node IN of $G^Z$ with the node $v_{in}$ and the node OUT of $G^Z$ with the node $v_{out}$, respectively. The united vertices will have the labels ***in*** and ***out***, respectively (see Fig. 4. and Fig. 6). Let the united graph be $G_k$.

The sub process finishes for a given non-terminal when all vertex-labels terminate in terminal symbols. The aim of the sub process is to construct a finite substitution tree. All leaf nodes need to be terminals.

Remark 1.

During processing, Alg. 1. can insert the sub graphs of all current nested attributes (if any) in a single step (Fig. 5. and Fig. 6).

# 3 NESTED RELATIONAL SCHEMA GRAPHS, RELATIONAL INSTANCES AND OPERATIONS

## 3.1 Nested Relational Schema Graphs and Schema Instances

We extend the definition of schema graph for regular expressions [1] to extended context-free languages. Using the notation of the ECFG from 2.3, we associate ordinary attribute names for the terminals T and nested attribute names for non terminals N. The extended language EL(G) consists of all paths of the generated graph by Alg. 2.

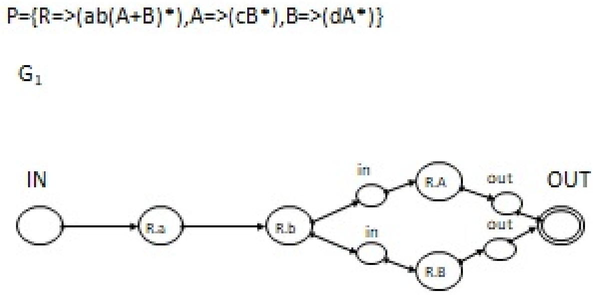


**Figure 2: Current generated graph in construction of a schema graph**

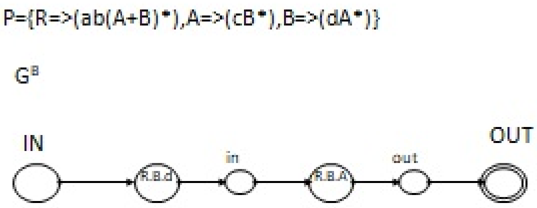


**Figure 3: Sub graph for a nested attribute in construction of a schema graph**

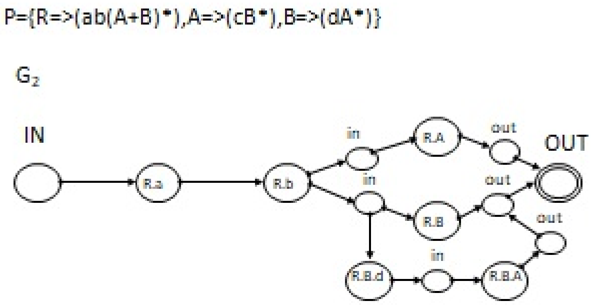


**Figure 4: Next generated graph in construction of a schema graph**

For each production rule A => $R_A$ the regular expression $R_A$ denotes a regular language $L_A \subseteq (N \cup T)^*$, the corresponding graph-representation can be constructed according to Alg. 1., using as input alphabet $\Sigma = N \cup T$. During derivation by the grammar G we substitute the non-terminal A by a sentence from $L_A$, so the vertex-labels set by Alg. 1. will be picked from either N or T. In order to demonstrate the path that leads to the attributes of a schema we use the dotted list of non-terminal symbols that will be used during the derivation process leading to a terminal symbol. The finishing (terminal) symbol is an ordinary attribute for the relation; the previous (non-terminal) symbols are the nested attributes. In the first step we use the start symbol R as the beginning (0-th)

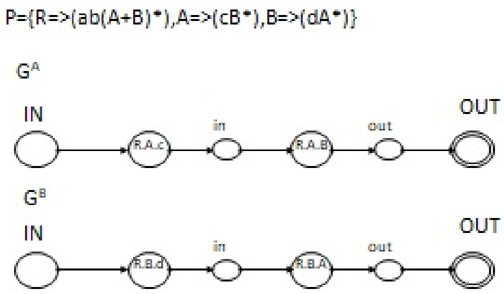


**Figure 5: Sub graphs for all current nested attributes in construction of a schema graph**

nested attribute. In order to create a schema graph for the extended context-free language L(G) generated by the grammar G we should repeat the construction for all vertices, labeled with a symbol X ∈ N. This serial of steps presents a given derivation choice of schemas. We can select either ordinary or nested schemas.

**Algorithm 3** Selection of a Schema from the Schema graph

```
Input: vertex labeled digraph SCH(G)=(V,E) representing the schema graph for an ECFG G=(N,T,R,P),
Output: a schema that may contain nested relational attributes, but their deepest schemas (leaf schemas) contain exclusively ordinary attributes
Step 1.
We select a traversal from SCH(G), which may contain non-terminal symbols. The non-terminals will be the nested attributes, the terminals the ordinary attributes. In an instance for the schema the ordinary attributes represent simple values the nested attributes take complex values (Def. 6.).
Step 2.
For each non-terminal we select a traversal from its graph, which gives the schema of the nested relation.
Step 3.
For all non-terminals in each relational schema we repeat Step 2.
Step 4.
The process finishes when all nested relational schemas contain exclusively terminals.
We call a generated scheme by Alg. 3. a legal schema.
```

**Example 1.** Let G ({R,A,B},{a,b,c,d},R,P) be an extended context-free grammar, where P={R=>(ab(A+B)*), A=>(cB*), B=>(dA*)}. There are the following possible generated (ordinary and nested) schemas:

R(abc), R(abd), R(abA[cd]), R(abA[cB[d]B[dcc])

R(abc) gives the ordinary schema R(a,b,c), in dotted form R(a,b,A.c)

The longest one is R(abA[cB[d]B[dcc]), generates the ordinary schema R(a,b,c,d,d,c,c), in dotted form R(a,b,A.c,A.B.d,A.B.d, A.B.A.c, A.B.A.c).

See the resulting graphs on Figures 2-7.

**Example 2**. Schema graph for the ECFG given in Example 1.

Let G ({R,A,B},{a,b,c,d},R,P) be an extended context-free grammar, the production rules are given in Fig. 8. Fig. 8. presents a schema graph for G. There are two sentences of the ECFL given by G represented in Fig. 8

(R.a, R.b, R.A.c, R.A.B.d)

(R.a, R.b, R.B.d, R.B.A.c)

The sequence of intermediate tuple-types:

Unnested case:

(R.a, R.b, R.A), . (R.a, R.b, R.A.c, R.A.B), ( R.a, R.b, R.A.B.d)

(R.a, R.b, R.B), (R.a, R.b, R.B.d, R.B.A.) (R.a, R.b, R.B.d, R.B.A.c)

Nested case for all nested attributes with {} set notation:

(R.a, R.b, R.A), . (R.a, R.b, R.A.{c,.B}), (R.a, R.b, R.A.{c,.B.{d}})

(R.a, R.b, R.B), (R.a, R.b, R.B.{d, A}), (R.a, R.b, R.B.{d, A.{c}})

## 3.2 Instances of Complex Relations with Context-free Schemas

Instances of relational databases with complex values consist of a finite number of complex relational schemas and a finite set of values of the sort given by each schema. (See [9] Chapter 20.) Following this, we define the instance of an extended relation with context free schemas.

For the given CFG we can suppose, without any loss of generality, that the language L(A) associated to any non-terminal symbol A is not empty. Using this assumption, any traversal generated by Alg. 2. is a legal schema instance. Legal schema instances are traversals accepted by Alg. 3. Let SCH(G) denote the set of schemas given by language L(G).

A schema instance from SCH(G) specifies a tuple-type. For a non-terminal symbol B in this sort a tuple-type is given from SCH(B). The associated value in a relation instance can be a simple tuple (unnested case) or a table as a set of tuples from this type (nested case).

**Definition 3**. A relation instance of the context-free schema SCH(G) is given by the pair (***R***,***I***), where ***R*** is a finite subset of SCH(G), and ***I*** is the set of complex valued relation instances for each element of ***R***. For r*ε**R*** the corresponding instance is denoted by ***I***(r).

**Example 3**.

Let G be the CFG from Example 1.

G={R=>(ab(A+B)*), A=>(cB*), B=>(dA*)}, An associated XML DTD fragment:

```
<!ELEMENT TABLE(R*)>
<!ELEMENT R (a,b,(A|B)*)>
<!ELEMENT A (c,B*))>
<!ELEMENT B (d, A*)>
<!ELEMENT a (#PCDATA )>
<!ELEMENT b (#PCDATA )>
<!ELEMENT c (#PCDATA )>
<!ELEMENT d (#PCDATA )>
```

A fragment of XML instance representing the longest schema from Example 1, is R(abA[cB[d]B[dcc]). It generates the ordinary schema R(a,b,c,d,d,c,c), in dotted form R(a,b,A.c,A.B.d,A.B.d, A.B.A.c, A.B.A.c).

Case of no nested type, the corresponding XML fragment:

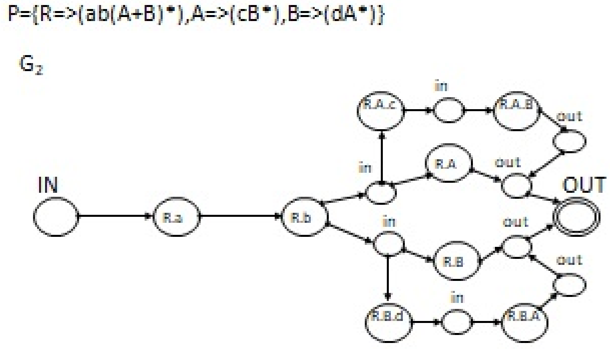


**Figure 6: Next generated graph in construction of a schema graph when inserting all current sub graphs**

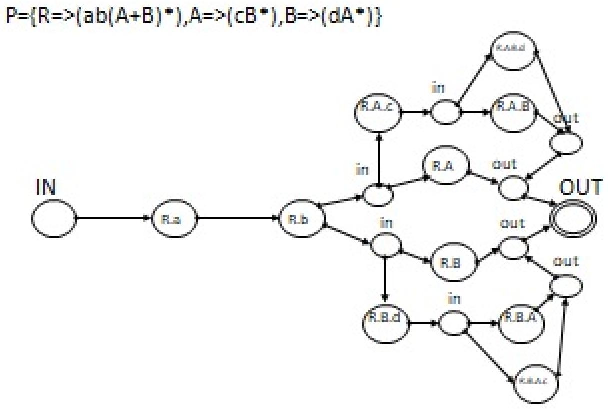


**Figure 7: Schema graph for the ECFG in Example 1.**

<R> <a>1</a> <b>2</b>
<A> <c>3</c>
<B> <d>4</d> </B>
<B> <d>5</d> <c>6</c> <c>7</c> </B>
</A>
</R>

Case of nested type; R(abA[c**B**[d]B[dcc]), the bold B specifies nested relation, the corresponding XML fragment:

<R> <a>1</a> <b>2</b>
<A> <c>3</c>
<**B**> <d>4</d> </**B**> <**B**> <d>8</d> </**B**> <**B**> <d>9</d> </**B**>
<B> <d>5</d> <c>6</c> <c>7</c> </B>
</A>
</R>

### 3.3 Operations on Complex Relations

Next we give some hint to define algebraic operators over SCH(G) instances. The details are left to the readers.

Set operation are defined over use two instances, (***$R_1$ , $I_1$***), and (***$R_2$ , $I_2$***). First, we take the operation over the schemas, union, intersect, minus of ***$R_1$ and $R_2$.*** Then for each pair $r_1$ *ε* ***$R_1$*** and $r_2$ *ε* ***$R_2$*** – where $r_1$ and $r_2$ are of same sort, we use the standard relational set operation .

Cross product can be defined over instances from two schemas generated by context-free grammars $G_1$ and $G_2$. Using the standard notation SCH($G_1$) X SCH($G_2$)= SCH($G_1G_2$), where $G_1G_2$ is the grammar of the concatenated languages. At the instance level, for

($\boldsymbol{R_1}$ , $\boldsymbol{I_1}$) , and ($\boldsymbol{R_2}$ , $\boldsymbol{I_2}$). form SCH($G_1$) and SCH($G_2$) respectively the cross product is taken for each pair each pair $r_1$ $\varepsilon$ $\boldsymbol{R_1}$ and $r_2$ $\varepsilon$ $\boldsymbol{R_2}$.

The most specific operation on nested relation is the pair Nest and Un-nest. These operations are defined on two instances $r_1$ $\varepsilon$ $\boldsymbol{R}$ and $r_2$ $\varepsilon$ $\boldsymbol{R}$. The traversals on the schema graph are the same for $r_1$ and $r_2$ but the choice of nested version or simple version of traversing the graph of a non-terminal B is different. See the exact definition these operation in [9] Chapter 20.2.

Usually, a projection operator result in an instance of a sort that does not belong to SCH(G) .It is possible to define a special projection operation when we allow for the role of some non-terminal to include the empty sentence. This means that a short cut of traversing a non-terminal vertex is possible. So, we can project out an attribute of this type and the result will remain inside SCH(G).

The situation with the natural join is similar, the result will not remain in SCH(G). It is possible to define a special language operator which associates a context-free language to the corresponding natural join operator on the instances of SCH(G). The common attribute list of instances $r_1$ $\varepsilon$ $\boldsymbol{R}$ and $r_2$ $\varepsilon$ $\boldsymbol{R}$. is given by the longest beginning common path of the traversals. Let the traversals be *xy* and *xz*. The result of the natural join is of sort given by *xyz*. This way we defined a new product-like operator on as schema graph, the self-natural join graph.

## 4 FUNCTIONAL AND KEY DEPENDENCIES ON SCHEMAS FOR FORMAL LANGUAGES

### 4.1 FD on Extended Context-free Languages with Restricted Grammar

Functional and key dependencies on ECFGs have been defined in [2]. Those definitions used a restricted ECFG allowing production rules of two forms only:

A => $R_A$, where A ∈ N, $R_A$ is a regular expression over N,

A => u, where A ∈ N, u ∈ T.

Moreover, the two sets of non-terminals those were LHS in group 1. or 2. were disjoint.

With these restrictions, the dependency definitions for regular languages could be applied for ECFGs too. Notice that the used schemas were ordinary: they contained no nested attributes. In the following we use ECFL/ECFG without the restrictions 1. and 2. above, applied in [2].

### 4.2 FD on Extended Context-free Languages using Nested Attributes

**Definition 4.** (Extended Relation for Extended Context-free Language). Let L be an ECFL and let G (G=(N,T,R,P)) be its generating grammar. We say that the set of terminal symbols T is a set of attribute names. Let $w=w_1 \dots w_n \in L$ a sentence, then we say that w is an extended relational tuple type over T. Let $dom_u$; u ∈ T be sets of data values, then $\{(w_1{:}a_1, \dots, w_n{:}a_n) \mid a_i \in dom_{wi}\}$ is the set of possible tuples of type w. A finite subset of these tuples is an instance of the extended relation. We say that the set of the tuple types for all w ∈ L compose the schema of an extended relation based on L. Each tuple type complies with an (IN,...,OUT) traversal on the schema graph SCH(G). The tuple types of all tuples in an instance compose the schema for the instance.

Let L be an ECFL and let G (G=(N,T,R,P)) be its generating grammar. In order to specify the (left and right) sides of a functional dependency we should pick up two sets of attributes (either ordinary – Def. 4. - or nested ones), one set for the left side (denoted by X), another one for the right side (denoted by Y). Using a schema from SCH(G) we can select ordinary attributes for both sides of an FD and we can define syntax and semantics for this FD similarly to the case of regular FD [2]. In the following, we deal with scoped FD on ECFL using nested attributes.

We can choose nodes visited by a traversing and state that each visiting of these nodes would be selected. We can choose starting and ending points for a path in the traversing, so that this pair of nodes will be selected at each closing of that path.

Let G=(N,T,S,P) be an ECFG. For each production rule, A => $R_A$ the regular expression $R_A$ denotes a regular language $L_A \subseteq (N \cup T)^*$, the corresponding graph-representation (denoted by G($R_A$)) can be constructed according to Alg. 1. During derivation by the grammar G we get the (not necessary regular) language generated from A: we denote L(A) ⊆ $T^*$ the strings derived from A by the rules in P.

Remark 2.

We can assume without loss of generality that each non-terminal symbol occurs once as LHS in a production rule, because the rules A => $R^1_A$ and A => $R^2_A$ can be replaced with the rule A => ($R^1_A$ + $R^2_A$) so that the obtained grammar is equivalent with the original one [10J.] .

Let t ∈ L(A), let U ∈ N be a non-terminal symbol so that U ∈ $R_A$. We can interpret U as attribute of A. U, as start symbol is the root of the ECFG $G_U$=G(N,T,U,P), so when generating t by G, some sentences of L(U) will be generated by the way, let these sentences be $u_1,\dots,u_k$ (k ≥ 1). Then t= $\omega_1 u_1 \dots \omega_k u_k \omega_{k+1}$, where $\omega_i \in T^*$, $1 \le i \le k+1$. We interpret the projection of U to t as $t[U]=u_1 \dots u_k$.

Definition 5. Let G=(N,T,S,P) be an ECFG, let $\propto \ \in (N \cup T)^*$ be a symbol string, then the language $L(\propto) \subseteq T^*$ is the set of all strings that can be derived from ∝ by the production rules in P for the non-terminals and letting each terminal symbol on its place. Formally, let $\propto = \propto_1 \propto_2 \dots \propto_n$, $\propto_i \ \in N \cup T, 1 \le i \le n$, then $L(\propto)=\{\omega \in T^* \mid \omega = \omega_1\omega_2 \dots \omega_n\}$, so that either $\alpha_i \overset{+}{\Rightarrow} \omega_i$, where $\overset{+}{\Rightarrow}$ is the transitive closure of the derivation, when $\alpha_i.$ ∈ N, or $\omega_i = \alpha_i$, *when* $\alpha_i \in T$.

According to Def. 5. we can assign values (taken from a non-empty domain set D) to the terminal symbols. Let u ∈ T be terminal symbol, then let $D_u \in D$ be a set so that when u, v ∈ T, u ≠ v then $D_u \cap D_v = \emptyset$.

The mapping *val*: $u \in T \rightarrow D_u$ assigns a domain value to a terminal symbol so, when the assignment will be made for a string of terminal symbols, then each assignment occurs autonomously, that is, *val*($u_1$) ≠ *val*($u_2$) can occur also when $u_1 = u_2$. Obviously, when $u_1 \neq u_2$ then *val*($u_1$) ≠ *val*($u_2$).

For ω ∈ L(A), let $\omega = \{u_1u_2 \dots u_n\}$ then *val*(ω) = $\{v_1v_2..v_n\}$ is a valuation of ω, where $v_i = val(u_i)$, $1 \le i \le n$.

**Definition 6.** (Complex valued tuple) [9]

Let G=(N,T,S,P) be an ECFG, let A ∈ N be a non-terminal symbol and let $\propto \in L_A$, $\propto = (A_1A_2 \dots A_n\}$, $A_i \in N \cup T$, $(1 \le i \le n)$

be a string of either non-terminal or terminal symbols. Let $\beta = (B_1, \ldots, B_k)$, $(0 \leq\ k \ll n)$ the string of non-terminals in $\propto$. We say that the $B_i$-s are nested attributes of $\propto$, which is a sentence format in the scope A. Let $\gamma = (C_1, \ldots, C_l)$, $(0 \leq\ l \ll n)$ the string of terminals in $\propto$. Obviously, k + l = n. Let $\omega \in L(\propto)$, $\omega =\{\omega_1\omega_2 \ldots \omega_n\}$ so that either $\omega_i \in L(A_i)$, $0 \leq i \leq n$, when $A_i \in N$, $(0 \leq i \leq n)$, or $\omega_i = A_i$, $0 \leq i \leq n$, when $A_i \in T$, $(0 \leq i \leq n)$. A valuation of $\omega$ is a complex valued tuple t of A, denoted by $t_{CV}=val_{CV}(\omega)$.

Regular functional dependencies are presented in [2] syntactically defined on the graph for the accepting FSA of a regular language, and semantics were given for them on sentences of the language. We extend this definition to ECFG with nested attributes so that the syntax of the FDs will be defined on a single regular expression (using one production step only), but for the semantics we use the whole derivation tree of the ECFG. With this restriction we can yet handle most real-life applications, meaning "horizontally" connected data, and it allows a quadratic complexity of implication.

Definition 7. (Assignment)

Let L be an ECFL and let G=(N,T,R,P) be its generating grammar. Let A ∈ N be a non-terminal symbol and let $G(R_A)$=(V,E) be the graph representation of $R_A$ (Alg. 1.). We say that the tuple Y= (Y1,Y2), where Y1 ⊆ V and Y2 is a sub graph of the transitive closure of $G(R_A)$ is an assignment on $G(R_A)$. Y1 is taken from the non-recurred part of $G(R_A)$, Y2 refers to nodes and edges whose are (could be) repeatedly visited during a traversing.

Let Y be an assignment, Y selects a unique subsequence from a given sentence format as follows:

Let G=(N,T,S,P) be an ECFG, let A ∈ N and let $G(R_A)$ be the corresponding graph representation. Let $w=\{v_1,v_2,...,v_n\}$ be a traversing on $G(R_A)$.

**Definition 8.** (Selection on Scope). Let Y= $(Y_1, Y_2)$ be an assignment on $G(R_A)$ for the scope A and let w be a traversing on $G(R_A)$. The symbols in $Y_1$ will be selected in order of their exploration (when visited). For each edge e ∈ $Y_2$ when the edge will be closed on the shortest path between its endpoints during the traversing on w, these two endpoints will be selected in their succession order (when visited at all). That is, if the two endpoints of the closing path are A and B ($A=_{vi}$,$B=v_j$ for some $1 \leq i \leq j \leq n$) then that path will be selected which does not contain neither A nor B. The nodes in $Y_2$ will be selected by each visiting (if any) during the traversing on walk (w). The selection will be processed for all edges and nodes in $Y_2$ autonomously. By the end of the selection the from w selected symbols build up the (possibly empty) array $w[Y]=(v_{i_1}, \ldots, v_{i_k})$ $(1 \leq i_1 < i_2 < \ldots < i_k \leq n$ $(k \gg 0)$.

Let w be a traversing on $G(R_A)$, let $\omega \in L(w)$, and let $t=val(\omega)$ be a tuple of A. We interpret the w[Y] sequence of symbols as set of "attributes" that projects the tuple *t* to the values $t[Y]=val(\omega [Y])$, that is, let $w=(v_1,v_2,...,v_n)$, let $w[Y]=( v_{i_1}, \ldots, v_{i_k})$ $(1 \leq i_1 < i_2 < \ldots < i_k \leq n$ $(k \gg 0)$ and let $\omega = \{\omega_1\omega_2 \ldots \omega_n\}$ then $t[Y]=val(\omega_{i_1})val(\omega_{i_2}) \ldots val(\omega_{i_k})$.

If w[Y]={}, then t[Y]={} as well.

Concerning the regular language $L_A$ we can define functional dependency over G $(R_A)$, considering the non-terminal A as the scope for the functional dependency (Def. 8.).

**Definition 9.** (Scoped Functional Dependency)

Let A be a scope in the ECFG G and let $G(R_A)$ be the corresponding graph representation. Let $X=(X_1,X_2)$ and $Y=(Y_1,Y_2)$ be two assignments (Def. 7.) over $M_A$. A functional dependency defined over $G(R_A)$ ($FD_A$) is an expression of the form X → Y. The R (finite) database instance of A satisfies the X → Y functional dependency (denoted by R |= X → Y), if for any two $t_1$,$t_2$ ∈ R tuples $t_1[X]=t_2[X]$ can be fulfilled only then, when $t_1[Y]=t_2[Y]$ also comes true. We call the case Y= $G(R_A)$ key dependency.

**Example 4**. The example XML instance (Fig. 8) describes data of course participant students, the corresponding DTD contains the following declaration:

<!ELEMENT Courses (Course+)>
<!ELEMENT Course (Cid,Std+))>
<!ELEMENT Std ((Stid,Stn,Stl)+)>

Based upon this declaration we can define the following two scoped functional dependencies (Def. 9.):

Scope Course: ({Cid},{}) → ({},{Stid,Stn,Stl}) : key dependency
Scope Std: ({Stn},{}) → ({Stl},{ })

## 5 CONCLUSION AND FUTURE WORKS

This paper presents schema graphs for extended context-free languages, based on the graph representation for the regular expressions and defines functional and key dependencies and set operations over them.

Our model offers the tools for a normal form of extended relations on ECFG, but to specify a normal form for extended relations on ECFG is a hard problem, because the set operations on schemas can lead out from the world of context free languages: the intersection of two ECFLs is not context-free. As a future work we could try to find a possibility to specify a normal form for extended relations on ECFG. Another possible theme to continue our work is to describe join dependencies for ECFG.

## REFERENCES

[1] A. Benczúr and G. I. Szabó. 2016. Towards a Normal Form and a Query Language for Extended Relations Defined by Regular Expressions. *Journal of Database Management*, vol. 27, no. 2, pp. 27-48.

[2] G. I. Szabó and A. Benczúr. 2012. Functional Dependencies on Extended Relations Defined by Regular Languages. *Springer*, LNCS, vol. 7153.

[3] A. Benczúr and G. I. Szabó. 2014. Towards a Normal Form for Extended Relations Defined by Regular Expressions. in *LNCS 8716*, ADBIS'14.

[4] M. Arenas and L. Libkin. 2004. A normal form for XML documents. *ACM TODS 29*, pp. 195-232.

[5] M. Vincent, J. Liu and C. Liu. 2004. Strong functional dependencies and their application. *ACM ToDS 29*, pp. 445-462.

[6] G. Berry and R. Sethi. 1986. From regular expressions to deterministic automata. *Theoretical Computer Science, Volume 48*, pp. 117-126.

[7] R. Alur, K. Etessami and M. Yannakakis. 2001. Analysis of recursive state machines. in *Springer*, International Conference on Computer Aided Verification.

[8] E. Orachev, I. Epelbaum, R. Azimov and S. Grigorev. 2020. Context-free path querying by kronecker product. in *LNCS 12245*, ADBIS 2020.

[9] S. Abiteboul, R. Hull and V. Vianu. 1995. Foundations of Databases, Addison-Wesley.

[10J.] Albert, D. Giammarresi and D. Wood. 2001. Normal form algorithms for extended context-free grammars. *Theoretical Computer Science, Volume 267, Issues 1–2*, pp. 35-47.

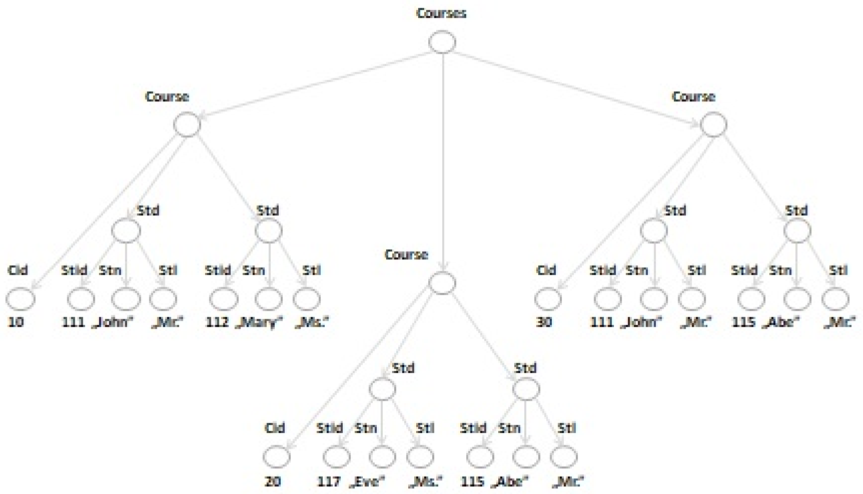


**Figure 8: XML instance for courses data**

# Synthetic Data Generation: A Comparative Study

Asha Mannarapotta Venugopal
University of Passau
Passau, Germany
asha.mannarapottavenugopal@uni-passau.de

Tung Son Tran
University of Passau
Passau, Germany
tungson.tran@uni-passau.de

Markus Endres
University of Passau
Passau, Germany
markus.endres@uni-passau.de

## ABSTRACT

Generating synthetic data similar to realistic data is a crucial task in data augmentation and data production. Due to the preservation of authentic data distribution, synthetic data provide concealment of sensitive information and therefore enable Big Data acquisition for model training without facing privacy challenges. Nevertheless, the obstacles arise starting with acquiring real-world open-source data to effectively synthesizing new samples as genuine as possible. In this paper, a comparative study is conducted by considering the efficacy of different generative models like *Generative Adversarial Network* (GAN), *Variational Autoencoder* (VAE), *Synthetic Minority Oversampling Technique* (SMOTE), *Data Synthesizer* (DS), *Synthetic Data Vault with Gaussian Copula* (SDV-G), *Conditional Generative Adversarial Networks* (SDV-GAN), and *SynthPop Non-Parametric* (SP-NP) approach to synthesize data with regard to various datasets. We used the pairwise correlation and Synthetic Data (SD) metrics as utility measures respectively between real data and generated data for evaluation. Accordingly, this paper investigates the effects of various data generation models, and the processing time of every model is included as one of the evaluation metrics.

## CCS CONCEPTS

• **General and reference** → *Surveys and overviews.*

## KEYWORDS

Synthetic Data, Neural Networks, Generative Models

**ACM Reference Format:**
Asha Mannarapotta Venugopal, Tung Son Tran, and Markus Endres. 2022. Synthetic Data Generation: A Comparative Study. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3548785.3548793

## 1 INTRODUCTION

Synthetic data refers to artificial information rather than those that are recorded from real-world events via direct measurement [7]. Artificial data is used when real data is not available, cannot be used due to privacy concerns and avoids business data vulnerable to data breaches.

Using real data for different purposes like algorithm testing, Machine Learning training or various Data Science applications in business and industries suffer from different problems: original data is often highly secure, takes a lot of time to be accessible, and it cannot be used for testing hypothetical scenarios. Therefore, generating synthetic data is quite important for researchers and business developers to overcome real data usage restrictions. It allows to simulate not yet encountered conditions and it can be generated to meet specific needs or conditions that are not available in existing (real) data. Another familiar constraint is the lack of specific characteristics in a dataset required for certain applications or domains, which typically cannot be obtained effortlessly and economically.

In practice, often scanty authentic data serve as template from which synthetic data can be produced algorithmically, e.g. when privacy requirements limit data usage and variability. Typical applications where synthetic data is a "must-have" are autonomous driving, financial services, healthcare, model training, and consumer behaviour evaluation in marketing and social media analysis.

Research communities and organizations involved in Machine Learning development need adequate datasets consisting of various characteristics for experimental purposes. Large, accessible and privacy compromising-free datasets therefore are much desired. Nevertheless in many cases, real data is often not sufficient to comprise such a dataset that meets the demand required to carry out experiments and approaches due to numerous concerns as aforementioned. Missing values or corrupted records due to errors in measurement or data encryption and storage, data is too expensive to acquire due to technological constraints or consent requirements, are among the many popular contributing reasons. Synthetic data henceforth becomes a promising alternative to alleviate these limitations and opens up opportunities in a wide range of domains like privacy protection, image generation, healthcare, data mining, pattern recognition, etc.

In this paper, we present a comparative study in order to identify the most efficient data generation method according to certain use cases. Seven models were inspected, namely *Generative Adversarial Network* (GAN), *Variational Autoencoder* (VAE), *Synthetic Minority Oversampling Technique* (SMOTE), *Data Synthesizer* (DS), *Synthetic Data Vault with Gaussian Copula* (SDV-G), *Conditional Generative Adversarial Networks* (SDV-GAN), and *SynthPop Non-Parametric* (SP-NP). For each use case, identical configurations such as working environment, hardware, training dataset (Adult Census Data, Airbnb Data, and Airlines Data) from open sources and programs are applied to every model to ensure fairness in performance evaluation, which considers processing time, generation speed and generated data quality. To the best of our knowledge, many of these models have been widely applied in contemplate

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548793

works, but there have not been a research that examines their efficacy in comparison and thus, this becomes the objective of our contribution.

This paper is structured as follows: Section 2 describes our used data generation models, the architecture pipeline, and the used datasets. Details on the implementation strategies and challenges can be found in Section 3. All experiments and results are presented in Section 4. In Section 5 we discuss related work, and Section 6 contains a summarization.

## 2 BACKGROUND

In this section we provide an outline of our synthetic data generation pipeline. We give a brief overview on our architecture, the data preprocessing step, the used models and datasets. We also present details of our evaluation techniques.

### 2.1 Pipeline Architecture

Our workflow is defined by four steps (Figure 1), starting with *data collection and preprocessing* where the preprocessed data becomes the input of the training phase, in which various features and characteristics such as data types, value ranges, pattern and distribution, etc. are extracted and captured. Data quality requirements include the consistency, accuracy, integrity, timeliness, interpretability, and believability of the dataset. The raw data, which is the original form of the real data supposed to be used as the training data, most often comprises incomplete, inconsistent data and lack of values or attributes depending on the different types of datasets. Therefore, we transformed the raw dataset into a clean and understandable dataset that can be used for training with the generator models [11]. In order to achieve this, we performed *Data Collection*, *Data Cleaning*, and *Data Analysis and Extraction. Data Collection* involves gathering and measuring the featured variables of the datasets enabling the target outcome for the next phases such implementation and evaluation. *Data Cleaning* is applied to check and fill in the missing values, remove noise data, detect or remove outliers, and correct inconsistent data. *Data Analysis and Extraction* takes place by reviewing the data and checking them for missing values, and removing noise data if any.

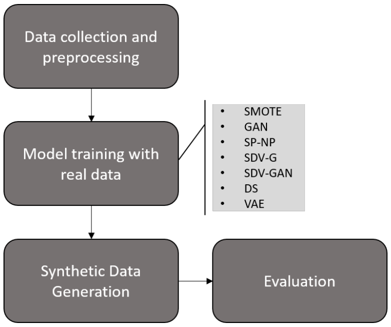

Data collection and preprocessing
Model training with real data
SMOTE
GAN
SP-NP
SDV-G
SDV-GAN
DS
VAE
Synthetic Data Generation
Evaluation

**Figure 1: Pipeline architecture**

Trained models subsequently can be readily used for inference tasks, where synthetic data can be generated based on user configurations queried to the models. Such configurations can specify desired sample quantity to be produced, value range as well as distribution. The data distribution present in the input schema (original data) is achieved to be similar as that of the output schema (generated data). Finally, the evaluation is conducted on the generated data to assess quality as well as to compare models performance according to certain evaluation metrics which will be further described subsequently.

### 2.2 Synthetic Data Generation Models

Seven frameworks considered in this study, namely *Synthetic Minority Oversampling Technique* (SMOTE), *Generative Adversarial Network* (GAN), *SynthPop Non-Parametric* (SP-NP), *Synthetic Data Vault with Gaussian Copula* (SDV-G), *Variational Autoencoder* (VAE), *Data Synthesizer* (DS) and *Conditional Generative Adversarial Networks* (SDV-GAN) are outlined in Table 1. A brief summary of each model is disclosed in Section 3, while their detailed descriptions are referred to the bibliography.

**Table 1: List of used models**

| Year | Name | Full name | Ref. |
|---|---|---|---|
| 2002 | SMOTE | Synthetic Minority Oversampling Technique | [2] |
| 2014 | GAN | Generative Adversarial Network | [6] |
| 2015 | SP-NP | SynthPop Non-Parametric | [13] |
| 2016 | SDV-G | Synthetic Data Vault with Gaussian Copula | [14] |
| 2017 | DS | Data Synthesizer | [16] |
| 2017 | VAE | Variational Autoencoder | [8] |
| 2019 | SDV-GAN | Conditional Generative Adversarial Networks | [17] |

### 2.3 Datasets

For our experiments we used datasets consisting of varying records. First, the *Adult Census Data*[1] (30,162 records with 11 attributes after data preprocessing) is the census data from 1994 based on income (originally possess 48,842 records with 14 attributes) with multivariate dataset characteristics consisting of both categorical and numerical data. Second, *Airbnb Data*[2] (213,451 records with 11 attributes after data preprocessing) has details of new users from Airbnb with demographics, web records and other statistics (initially with same number of records and 16 attributes) holding different type of attributes that consists of categorical, numerical and timeseries data. And third, *Airlines Dataset*[3] (1,046,595 records after data preprocessing) has flight travel details of adult passengers (1,048,575 records with 30 attributes before data preprocessing) with several attributes consisting of categorical and numerical data. We have choosen these datasets because of different type of attributes holding varying data distribution and enabling the betterment of implementation process in finding the advantages and disadvantages of each model while handling the datasets.

### 2.4 Evaluation Techniques

This section describes the two evaluation techniques used to evaluate the datasets on the compared models on the basis of it efficacy and utility. The used techniques are *Pairwise Correlation* and *SD Metrics*.

*2.4.1 Pairwise Correlation and Proximity Value.* Pairwise correlation is the correlation distance observed between two variables. Correlation distance is a famous method of estimating the distance

[1] UCI Machine Learning Repository (2019): https://archive.ics.uci.edu/ml/datasets/census+income

[2] Airbnb: https://www.kaggle.com/competitions/airbnb-recruiting-new-user-bookings/overview

[3] AirlinesCodrnaAdult: https://www.openml.org/d/1240

between two random variables with limited variances. When the distance of two items is zero, at that point they are the equivalent, and vice versa [1]. The method evaluates the correlation of real data and synthetic data. The concept behind this technique is to determine if the relationship between the variables in the real data are preserved in the generated synthetic data, which is done as follows:

(1) Firstly by observing the $n$ variables of real data and storing it in a matrix `df_real.corr`.
(2) Then the $n$ variables of synthetic data are observed and stored in a matrix `df_synth.corr`.

By exploring the output values from data generation, we check whether the pairwise correlation structure of the real data is firmly reflected by the correlation structure of the synthetic data and then determine the pairwise correlation distance for both real data and synthetic data and calculate the difference between them. To accomplish this, the correlation difference `diff` is calculated as follows:

$$\texttt{diff = df\_real.corr - df\_synth.corr} \tag{1}$$

For a generated dataset of good quality, diff should be close to '0', which is the **proximity value**. The *proximity values* of bad quality are far from '0'. Proximity is the measure of similarity and dissimilarity between the data points. The proximity value referred in this case mean the value obtained after the observation of correlation distance between real data and synthetic data [5].

*2.4.2 SD Metrics.* SD metrics are developed under the SDV project [4] [14] both to evaluate the relationship of synthetic data with real data and to test the quality of the former one. It is a dataset-agnostic tool that supports various data modalities such as single column, column pairs, single table, multi table and timeseries. It also includes a variety of metrics such as statistical, detection, efficacy, privacy, Bayesian Network and Gaussian Mixture. These metrics are a combination of several metrics such as CSTest (Chi-Squared), KSTest (Inverted Kolmogorov-Smirnov D statistic), KSTestExtended, LogisticDetection (Logistic Regression Detection), SVCDetection (Support Vector Classification Detection), BNLikelihood (Bayesian Network Likelihood), BNLogLikelihood (Bayesian Network Log Likelihood), LogisticParentChildDetection (Logistic Regression Detection), and SVCParentChildDetection (Support Vector Classification Detection). Here, values close to '0' mean that the data is not of good quality, and values close to '1' mean the data is of good quality. SD Metrics is available as a library in **Python**.

## 3 IMPLEMENTATION DETAILS AND CHALLENGES

In this section we present more details on the used models and also discuss implementation issues and challenges we had so solve in order to compare all models.

### 3.1 SMOTE

SMOTE [2] is an over-sampling technique that generates data for minority classes by inputting a sample from each of such classes. Then it creates synthetic samples based on its corresponding k Nearest Neighbors (kNN) in the feature space. The new samples are generated by multiplying the difference between the input feature vector and that of a selected nearest neighbor with a random value between 0 and 1, then add the result to the input feature vector. This results in the decision boundary of the minority classes becoming more general.

*Implementation.* We adopted the SMOTE technique with *Label Encoding* and *Logistic Regression*. Label Encoding is used in converting the labels (data in each row or column or cell) into numeric format making it easier for the Machine Learning models to process the dataset whereas Logistic Regression helps in predicting the variables which are categorically dependent. In order to generate duplicate/fake values, SMOTE requires to have one or two columns. This allows the data to be manipulated and to produce new random values. SMOTE is initialized for oversampling so that the data can be transformed. This is done with a random state of *2* and also by setting a *sampling_strategy* ratio to assign the number of samples to be generated because the data was earlier split into two forms, one with one or two columns and reference and the other set that includes all the features of all the columns.

*Challenges.* SMOTE faces certain challenges while generating synthetic data. The main challenges include overlapping of classes, creating additional noise and it is also hard to implement it for very high dimensional data. We overcome this by analyzing the samples and their respective labels from the results of Label Encoding. The hyperparameters were standard to all the 3 datasets where we have the random_state as 2 and k_neighbours as 1. By tuning the hyperparameters in random_state and k_neighbours, the similarity or differences in resulting values were very minute. SMOTE in general requires sufficient amount of samples to compare during the oversampling phase, we must ensure there are sufficient samples (specifically where we have more than 2 labels) in the columns (particular column attributes) of the dataset. This was done during the data preprocessing where we analyze the data and check if all the attributes hold sufficient number of samples possessing more than 2 labels.

### 3.2 GAN

Generative Adversarial Network (GAN) [6] is a Neural Network (NN)-based approach consisting of two components: a generative model $G$ that learns the distribution from training data, then produces samples from it, and a discriminative model $D$ that tries to distinguish the generated samples as real or fake. The objective of the training process is to maximize the error of the discriminator $D$ on classifying samples produced by the generator $G$, thus inherently maximizing the synthesized samples similarity to authentic samples. Both models are trained independently, in which the generator gain more efficiency at producing similar (better) synthetic data, while the discriminator becomes increasingly skilled at identifying these data.

*Implementation.* The implementation of GAN mainly uses Label Encoding (similar like Synthetic Minority Oversampling Technique (SMOTE)) and Clustering. Clustering helps in dividing the data points into groups where the data points of the same group are similar to those in the other group. The batch size is set to 100 so that the whole set of data is handled in batches of 100 on each iteration. Next, two variables are initialized, one for the discriminator and the other for the generator with the value 1. It is mainly the number of discriminator network updates per adversarial training step and

[4] SD Metrics: https://github.com/sdv-dev/SDMetrics

number of generator network updates per adversarial training step, respectively. These act as the key hyperparameter arguments which are standard for all the datasets. Then the column labels and column data are initialized with the columns of the trained data for the class and if not for the column labels. The training data with no label is initialized with the training data of the data columns.

*Challenges.* In general, it is difficult to train GANs as they have several problems such as *non-convergence* and *mode collapse. Non-convergence* is when the parameters are unstable and do not converge. *Mode collapse* is when the generator collapses and generates limited variety of samples. There occurs an unbalance between generator and discriminator that causes overfitting and when it comes to hyperparameter selection, they become highly sensitive. To overcome this, it is important to balance between generator and discriminator to avoid overfitting in the training.

## 3.3 SP-NP

SynthPop Non-Parametric (SP-NP) [13] is a Python package of the original **R** package *SYNTHPOP* which uses the functions and structure of the mice multiple imputation package and extends it for the purpose of synthetic data generation. The basic functionality of *SYNTHPOP* is to generate synthetic forms of microdata containing sensitive information. It has two modes, namely parametric and non-parametric. Parametric is a mixture of logistic regression with linear regression, which uses binary, numeric, ordered and unordered factor of data types designated in a vector (default parametric method) and can be customized if needed. Non-parametric uses Classification and Regression Trees (CART) (default non-parametric method) that is based on classification and regression trees, which handles all types of variables having predictors and can be applied for all data types. In this paper, we use only the non-parametric version.

*Implementation.* The implementation of SP-NP involves Label Encoding, similar to SMOTE where the input data is encoded and given as input to the generator function. In general, SYNTHPOP requires dtypes (data types) as one of its main hyperparameters apart from the feature contents. So it is necessary to specify the data types of every column attribute in this dtypes respectively for each dataset. The results obtained from the generator is decoded to the original input data structure which serves to be our synthetic data.

*Challenges.* SYNTHPOP had to be customized for every single dataset based on their attribute type. It also does not work well with float and categorical data formats. This is the reason why we use Label Encoding to overcome this challenge.

## 3.4 SDV-G

Synthetic Data Vault with Gaussian Copula (SDV-G) [14] is a framework that generates synthetic data by utilizing a multivariate model derived from the intersection of various tables in a relational database. The modeling of the whole database is performed by taking input consisting of the data tables themselves and their corresponding metadata. The relationship between tables is computed recursively using Conditional Parameter Aggregation (CPA) to handle foreign key relations, while the relationship between columns of a table is computed using multivariate Gaussian Copula to calculate covariance. The framework is capable of performing model-based and knowledge based synthesis, where the former allows user to synthesize a complete database using solely the computed model, while the latter allows the completion of data based on some given information.

*Implementation.* The framework of SDV is advantageous for direct implementation for all types of dataset. Input and generated synthetic data are in the form of dataframes. It is also to be noted that Label Encoding can be used for all the SDV models. SDV-G is implemented directly by only assigning the Gaussian Copula model for the generator.

*Challenges.* In general, Gaussian Copula works only for linear correlation problems and not when there is tail dependence. Tail dependence is a concept of clustering of large events which are extremely important for risk management problems. So, for this task SDV-G servers to beneficial, but for clustering problems it is in question.

## 3.5 SDV-GAN

Conditional Generative Adversarial Networks (SDV-GAN) [17] is an adapted framework of SDV-G using Conditional GAN (CTGAN), a data generator to generate synthetic tabular (categorical) data based on GAN, which can handle varied feature types for both discrete and continuous data that claim to outperform the existing GAN models, Variational Autoencoder (VAE) and Bayesian Networks when applied on benchmarking datasets. This also includes some new techniques such as *mode-specific normalization* to augment the training process, change in architecture and to resolve the data imbalance by implementing *conditional generator* and *training by sampling*. We call this model as SDV-GAN since CTGAN is also a part of the SDV-G framework.

*Implementation.* Since the framework of SDV can be directly implemented for all types of datasets where the input data is in the form of a dataframe and the generated synthetic data is also in the form of a dataframe, the setup is similar to the implementation of SDV-G. So, SDV-GAN is also implemented directly by only assigning the CTGAN model for the generator.

*Challenges.* SDVs also have problems with producing complete results when the samples are too less for an attribute type. Larger datasets with sufficient amount of categorical data can overcome this issues and also smaller datasets, when trained multiple times, overcome this in certain cases.

## 3.6 VAE

VAE [8] is a NN-based approach capable of capturing complicated data distribution and produce synthetic data that are similar to the original data, which consists of two components: an encoder and a decoder. The former captures dimensional dependencies by mapping the input data into the latent space, while the latter takes the latent variable as input to generate samples. Mathematically, VAE is classified as a variational Bayesian method and thus is different in formulation to autoencoders despite the architecture similarity.

*Implementation.* The implementation of VAE is as difficult as GAN, escpecially for the purpose of generating categorical data. Here, we use Label Encoding just like SMOTE, GAN and SP-NP. Since VAEs already possess an encoder and decoder, the process of generation takes more time compared to any other model. But it is not possible to avoid the label encoding because the encoder and decoder functions result in some null values when implemented directly.

*Challenges.* The main challenge of VAE is the consumption of processing time. In future, we might be able to overcome this when implementing and running on a GPU environment.

## 3.7 DS

Data Synthesizer (DS) [16] is a framework consisting of three modules: a *Data Describer*, a *Data Generator* and a *Model Inspector*. The Data Describer inspects the input data types, correlations and attributes distribution to create a summary, from which synthetic samples are generated by the Data Generator. The Model Inspector is responsible for visualizing the summary, allowing accuracy evaluation and parameters adjustment. It has three modes: the default *correlated attribute mode*, the *independent attribute mode* for the cases of expensive correlation computation cost or inadequate samples, and the *random mode* for cases of exceedingly sensitive data.

*Implementation.* The framework of DS is implemented directly using the random mode as we need our data to be handled sensitively, but it has problems in generating timeseries data. We included the timeseries generation by using the datetime Python library. The main implementation is carried out by inputting the dataset file as a whole and drawing its features into a JSON file which acts a medium in generating the synthetic data having the same features as the real data. This is one of the reasons why it is time efficient. The threshold value is the hyperparameter for Data Describer which is standard for all the datasets and it is noted that this threshold is always less than the domain size when the attribute is categorical.

*Challenges.* Inability to generate timeseries is a major drawback to the model. But if this can be fixed using some additional timeseries generation methods, then this serves to be one very beneficial for synthetic data generation as the time consumption is very less.

# 4 EXPERIMENTS AND RESULTS

In this section, we present our comparative study and discuss the experiments conducted elaborately by describing and presenting the results obtained throughout the experimental phase [5].

## 4.1 Experimental Setup

The experiments are fourfold to the goals of this paper after successful completion of the implementation and evaluation. The experiments are mainly carried out on different datasets stored in CSV files, namely Adult Census Data (30,162 records), Airbnb Data (213,451 records) and Airlines Data (1,046,595 records). For more details we refer to Section 2.3.

The environmental setup for the experiment consists of Python 3.9, Jupyter Notebook with Anaconda3, PyCharm IDE, and required libraries installed on a Ubuntu platform. The Ubuntu platform is a server with Ubuntu 20.04, 754GB RAM, Intel(R) Xeon(R) Gold processor and CPU at 2.60GHz. The Data Preprocessing is done using Jupyter Notebook, Synthetic Data Generation and Evaluations are carried out on PyCharm.

## 4.2 Accumulated Results

The results obtained from the experiments are depicted in Tables 2 – 5 as well as in the form of heatmaps (Figures 2 – 4 in the Appendix). Finding the proximity and SD Metrics (cp. Section 2.4) is evaluated for each column of synthetic data against each column of real data. The models have good proximity, where '0' means we have a closer proximity and any value far from '0' means the proximity is low. Also we have a good SD Metric score, where '1' means the generated synthetic data is of good quality whereas '0' means the quality of data is low. The heatmaps represent the pairwise correlation distance between each column attribute of the synthetic data and each column attribute of the real data.

We performed different experiments on different datasets as described below:

1. The first experiment begins with generating and evaluating the *Adult Census Data* with 10K records for the 7 models SMOTE, GAN, SP-NP, SDV-G, SDV-GAN, VAE and DS.
2. The second experiment is with *full Adult Census Data* where we have 30,162 records for which we perform experiments for the models SMOTE, GAN, SP-NP, SDV-G, SDV-GAN and DS. We excluded VAE here because it took longer runtime of about 4 days and resulting in a value error occurred from the float point matrix of the generated dataset.
3. The fourth experiment is for *Airbnb* Data (213,451 records) which includes all necessary types of attributes such as categorical, numerical and timeseries. Here, we performed experiments for the models SMOTE, SP-NP, SDV-G and DS and not for GAN, SDV-GAN and VAE, because the models produce memory error due to huge number of records.
4. Finally, the last experiment is conducted on *Airlines Data* (1,046,595), which possess more than a million records on models SMOTE, SP-NP, SDV-G and DS. Similar to the Airbnb Data, we do not perform experiments on GAN, SDV-GAN and VAE.

*4.2.1 Results on Adult Data.* The first experiment with 10K records of Adult Census data is conducted and shows that the models SMOTE, GAN, SP-NP, SDV-G, SDV-GAN, VAE and DS are successfully implemented. From the represented results in Table 2, it is feasible to understand the performance nature and the comparison of the proximity levels and SD Metrics.

**Table 2: Adult Census Data (10K records)**

| Adult Census Data (10K records) | | | |
|---|---|---|---|
| | Proximity Level | SD Metrics | Processing Time |
| GAN | 0.0127 | 0.8564 | 26.2 minutes |
| VAE | 0.1903 | 0.3704 | 21.2 hours |
| SMOTE | 0.0055 | 0.7833 | 2.6 minutes |
| DS | 0.0190 | 0.0967 | 0.4 seconds |
| SDV-G | 0.0002 | 0.6434 | 7.0 seconds |
| SDV-GAN | 0.0065 | 0.6830 | 3.8 minutes |
| SP-NP | 0.0007 | 0.8595 | 2.6 minutes |

Table 3 shows the results conducted on the Adult Census Data with 30,162 records, whereas the heatmaps (Appendix, Figure 2) represent the pairwise correlation distance of synthetic data and real data. The models SDV-G and SP-NP have better scores with respect to the proximity level, while SMOTE, GAN, SP-NP and SDV-GAN have better scores with respect to SD Metrics and in terms of processing time, DS and SDV-G outperform the other models, but SMOTE and SP-NP have resulted in a reasonable processing time.

[5] Source Code: https://github.com/ashamvenu/SyntheticData.git

**Table 3: Adult Census Data Results**

| Adult Census Data | | | |
|---|---|---|---|
| | Proximity Level | SD Metrics | Processing Time |
| GAN | 0.0402 | 0.8621 | 1.9 hours |
| SMOTE | 0.0372 | 0.7296 | 8.9 minutes |
| DS | 0.0219 | 0.0977 | 1.3 seconds |
| SDV-G | 0.0029 | 0.6426 | 17.9 seconds |
| SDV-GAN | 0.0100 | 0.7134 | 25.7 minutes |
| SP-NP | 0.0054 | 0.8644 | 9.0 minutes |

*4.2.2 Results on Airbnb Data.* We carried out the experiment on the Airbnb Data. The models of GAN, VAE and SDV-GAN failed to run, producing a termination of the program due to huge amount of data samples ultimately causing a memory error. Hence, this experiment was performed only on SMOTE, SP-NP, SDV-G and DS. The results (Table 4) obtained from Airbnb data show that SMOTE and SP-NP result in better proximity compared to DS and SDV-G. In terms of processing time, DS scores far better than the other three models. The heatmaps found in the Appendix, Figure 3, show the correlation distance between the real data and synthetic data for each column and feature in the Airbnb Data. Table 4 depicts the results in terms of proximity level and processing time of Airbnb Data. SD Metrics is not evaluated for Airbnb Data as the SD Metrics is not capable for large datasets and result in memory error as SD Metrics framework is designed to evaluate the dataframes of synthetic data and real data as a whole.

**Table 4: Airbnb Data Results**

| Airbnb Data | | |
|---|---|---|
| | Proximity Level | Processing Time |
| SMOTE | 0.0036 | 1.5 hours |
| DS | 0.0093 | 33.6 seconds |
| SDV-G | 0.0102 | 2.6 minutes |
| SP-NP | 0.0008 | 1.6 hours |

*4.2.3 Results on Airline Data.* Similarly to Airbnb Data, the experiments are carried out only on SMOTE, SP-NP, SDV-G and DS models. The results (cp. Table 5) show that SMOTE and SP-NP result in better proximity compared to DS and SDV-G. Also here, the processing time of DS is far better than the other three models. Again, the heatmaps are shown in Figure 4 and present the correlation distance between the real data and synthetic data for each column in the Airlines dataset. Table 5 depicts the results in terms of proximity level and processing time of Airlines Data. Similar to Airbnb Data, we did not evaluate SD Metrics due to memory errors.

**Table 5: Airlines Data Results**

| Airlines Data | | |
|---|---|---|
| | Proximity Level | Processing Time |
| SMOTE | 0.0148 | 7.7 hours |
| DS | 0.0287 | 90.3 seconds |
| SDV-G | 0.0282 | 11.9 minutes |
| SP-NP | 0.0008 | 6.9 hours |

# 5 RELATED WORK

Synthetic data generation has been a significant research topic for the past two decades. However, most of the research work is based on artificial image generation and the focus on text data has been rising only in the recent years. There are many papers dealing with synthetic data generation and we are not able to mention all of them. In this section, beside the models proposed in Section 2 and 3, we mention the most relevant work related to our study.

In a comparative work conducted by Dandekar et al. [4] on Linear Regression, Decision Tree, Random Forest and Neural Networks, the results show that NNs are more effective w.r.t. utility and privacy on the basis of running time. However, the evaluation also shows that the normalized Kullback–Leibler divergence scores are more or less the same for all the four models. Considering the importance of implementing flexible models for synthetic data generation by calculating nuances of multivariate structures, Manrique-Vallier and Hu [12] proposed a Bayesian non-parametric method to preserve complex multivariate data relationships between different variables subject to structured zeros by a tool called Truncated non-Parametric Latent Class Model (TNPLCM) using Full Conditional Specification (FCS) approach, CART method and Random Forests. This results in producing high quality of analytical data which exposes low risk. For this reason, we have used SP-NP in our research, which uses CART method for its non-parametric version.

Peng and Telle [15] published a complete tool for synthetic data generation, employing three algorithms namely Random Data Generation, Decision Tree and Multilinear Regression for different use cases depending on their respective data mining pattern. While being satisfactory on the aspect of software functionality, adequate evaluation of output quality and processing time was not provided.

Beside textual data, synthetic time series generation is also in demand, particularly in domains where sensor data analysis is involved such as healthcare applications. Dahmen and Cook [3] introduced a Machine Learning approach based on hidden Markov models and models. The generated data have been evaluated using time series distance measures against the authentic, manually annotated smart-home data and exhibited to be highly realistic as well as capable of improving the accuracy of the model it was used to train on. However, the results show only the accuracy of which we find the scores to be not satisfactory considering the quality of the generated data in comparison with the real data.

Lecanzo and Arias [10] introduced three different generation methods with traditional datasets based on item-based generative models, where two of the models Itemset Generating Model (IGM) and Interesting Itemset Miner (IIM) are over itemsets and the third one Latent Dirichlet Allocation (LDA) is using textual corpora. Evaluation is carried out based on characteristics (pattern similarity), preservation of itemsets, privacy and runtime. This determines the strength and weakness of each model in which IGM has the lowest learning phase runtime, while IIM scores best in data generation. Though the results depicts the runtime in seconds, the amount of data used is unclear to consider the evaluation.

Leduc and Grislain [9] proposed an architecture called Composable Generative Model (CGM) which is an auto-regressive model that inputs column embeddings through a transformer, evaluated using Synthetic Data Gym (SDGym) benchmark and claiming CGM to be the best state-of-the-art approach. The concept of using an encoder, decoder and loss function seems reliable but the evaluation scores do not justify the claims with respect to synthetic data.

## 6 CONCLUSION AND DISCUSSION

In this paper we considered the challenge of generating synthetic data based on real data. The objective of our paper was to compare several state-of-the-art approaches to synthesize data in order to find out the efficiency of the proposed models. We used different real world datasets as a basis for our data-driven tasks, namely Adult Census data, Airbnb data, and Airlines data to analyze and depict the difference in performance as well as the behaviour of all seven models. We conducted several experiments of which the initial experiments show that SMOTE, SP-NP, SDV-G and DS are better among the 7 models chosen in terms of proximity, SD Metrics and processing time. GAN, SDV-GAN and VAE are not capable for large datasets, hence we run further experiments on SMOTE, SP-NP, SDV-G and DS to analyze which among these 4 models give us more promising results. By the end of all the experiments, we come to a conclusion that SMOTE and SP-NP are the most effective methods in terms of proximity and SDV-G and DS are effective in terms of processing time.

Our next step will be to consider models for text data generation, which is out of scope in this work due to being a more challenging task. Unlike the categorical data types, natural language generation involves language modelling which is a complicated and demanding process that requires more sophisticated model architectures, enormous amount of data, robust hardware to accommodate and longer training time. In addition, evaluating generated text quality is also an obstacle to be addressed, as identifying suitable metrics for automatic validation can be much less straightforward and counter intuitive, while human evaluation is certainly way too expensive to be included.

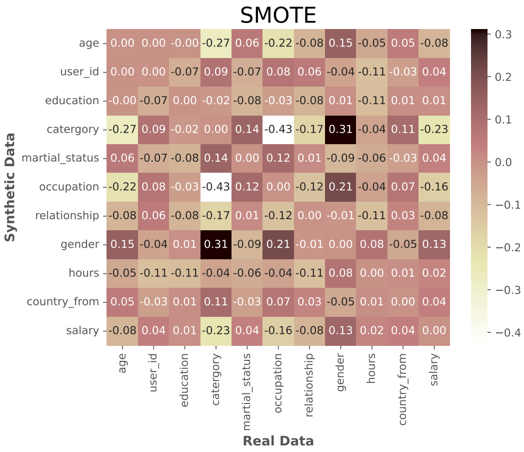
## ACKNOWLEDGMENTS

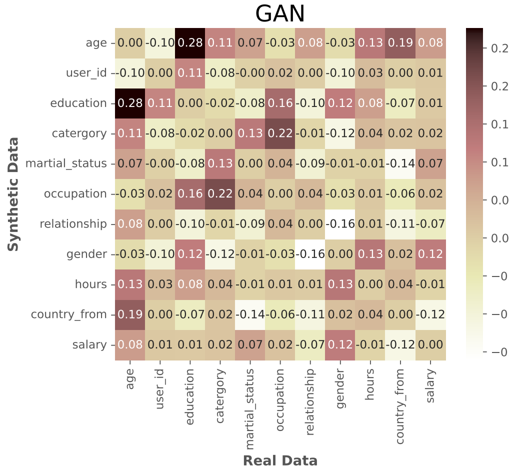
This research is part of the project "BigScience", which is funded by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy under the grant number DIK0259/01.

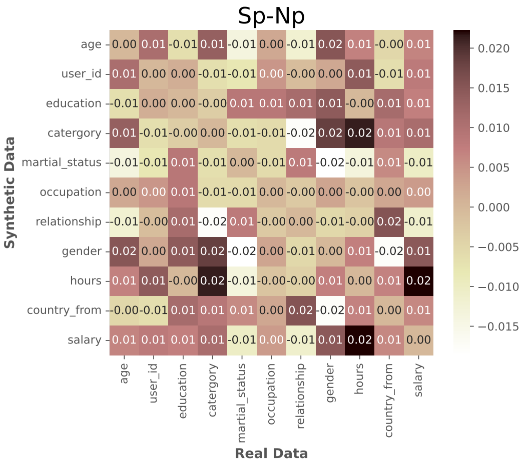
## REFERENCES

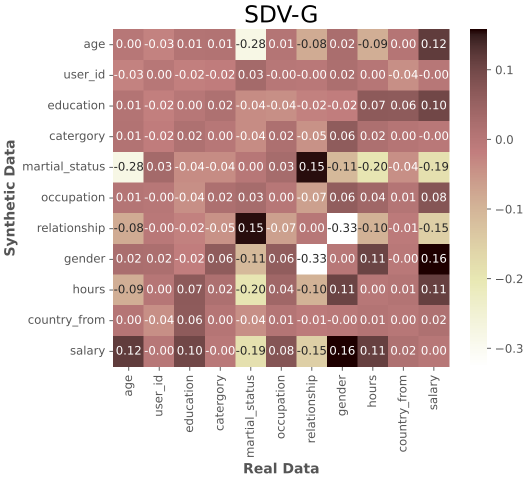
[1] Atanu Bhattacharjee. 2014. Distance Correlation Coefficient: An Application with Bayesian Approach in Clinical Data Analysis. *Journal of Modern Applied Statistical Methods* 13, 1 (2014), 23.
[2] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. 2011. SMOTE: Synthetic Minority Over-sampling Technique. *CoRR* abs/1106.1813 (2011).
[3] Jessamyn Dahmen and Diane Cook. 2019. SynSys: A Synthetic Data Generation System for Healthcare Applications. *Sensors* 19, 5 (2019). https://doi.org/10.3390/s19051181
[4] Ashish Dandekar, Remmy A. M. Zen, and Stéphane Bressan. 2018. A Comparative Study of Synthetic Dataset Generation Techniques. In *Database and Expert Systems Applications*. Springer International Publishing, 387–395.
[5] B.S Everitt and David C. Howell. 2005. *Encyclopedia of Statistics in Behavioral Science*. Vol. 3. Wiley, 1621–1628.
[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. *Advances in neural information processing systems* 27 (2014).
[7] McGraw Hill and S.P. Parker. 2003. *McGraw-Hill Dictionary of Scientific and Technical Terms*. McGraw-Hill Education. https://books.google.de/books?id=xOPzO5HVFfEC
[8] Diederik P Kingma and Max Welling. [n.d.]. Auto-Encoding Variational Bayes. https://doi.org/10.48550/ARXIV.1312.6114
[9] Johan Leduc and Nicolas Grislain. 2021. Composable Generative Models. *CoRR* abs/2102.09249 (2021). arXiv:2102.09249 https://arxiv.org/abs/2102.09249
[10] Christian Lezcano and Marta Arias. 2020. Synthetic Dataset Generation with Itemset-Based Generative Models. *CoRR* abs/2007.06300 (2020). arXiv:2007.06300 https://arxiv.org/abs/2007.06300
[11] Jasdeep Singh Malik, Prachi Goyal, and Mr. Akhilesh K Sharma. 2010. A Comprehensive Approach Towards Data Preprocessing Techniques & Association Rules.
[12] Daniel Manrique-Vallier and Jingchen (Monika) Hu. 2018. Bayesian Non-parametric Generation of Fully Synthetic Multivariate Categorical Data in the Presence of Structural Zeros. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 181 (02 2018). https://doi.org/10.1111/rssa.12352
[13] Beata Nowok. 2015. synthpop : An R package for generating synthetic versions of sensitive microdata for statistical disclosure control.
[14] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. The Synthetic Data Vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 399–410. https://doi.org/10.1109/DSAA.2016.49
[15] Taoxin Peng and Alexander Telle. 2018. A Tool for Generating Synthetic Data. In *Proceedings of the First International Conference on Data Science, E-Learning and Information Systems*. Association for Computing Machinery, New York, NY, USA, Article 22, 6 pages. https://doi.org/10.1145/3279996.3280018
[16] Haoyue Ping, Julia Stoyanovich, and Bill Howe. 2017. DataSynthesizer: Privacy-Preserving Synthetic Datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. Article 42, 5 pages. https://doi.org/10.1145/3085504.3091117
[17] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling Tabular data using Conditional GAN. *CoRR* abs/1907.00503 (2019). arXiv:1907.00503 http://arxiv.org/abs/1907.00503

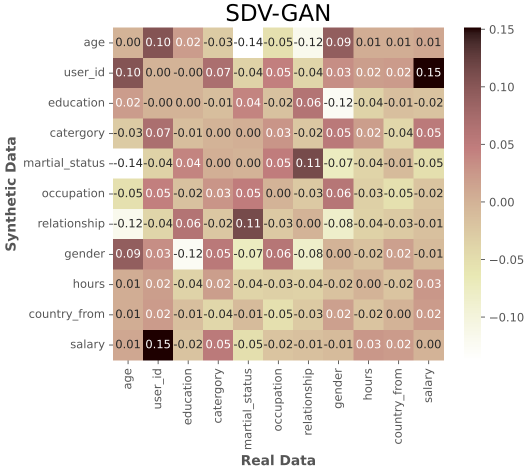
## APPENDIX











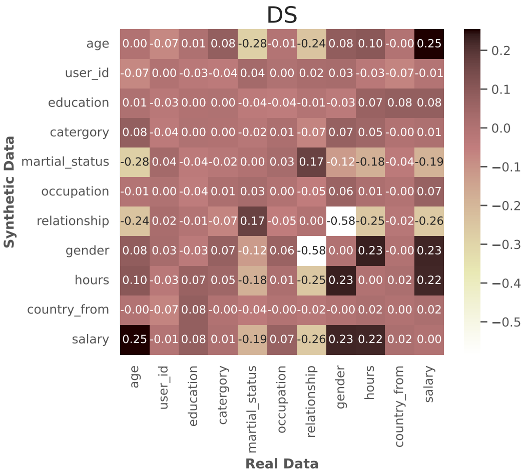
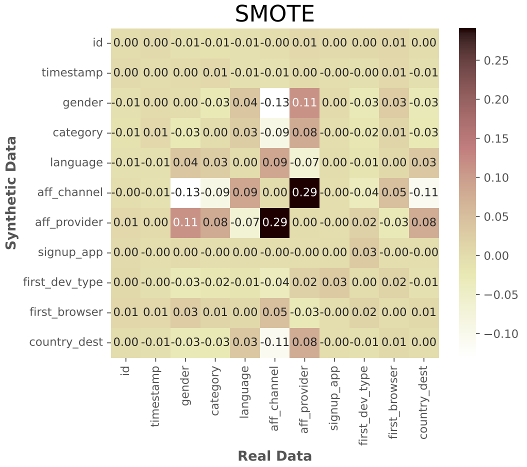
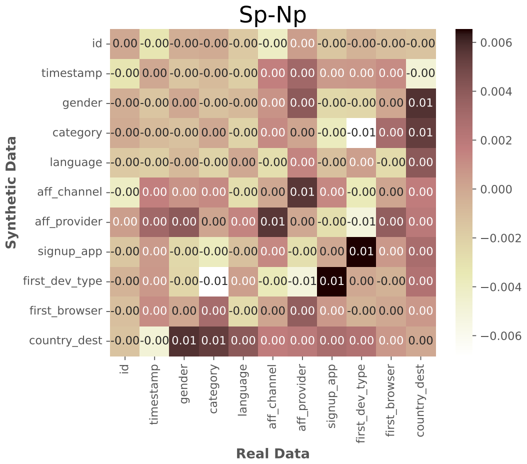


**Figure 2: Heatmaps of Adult Data**





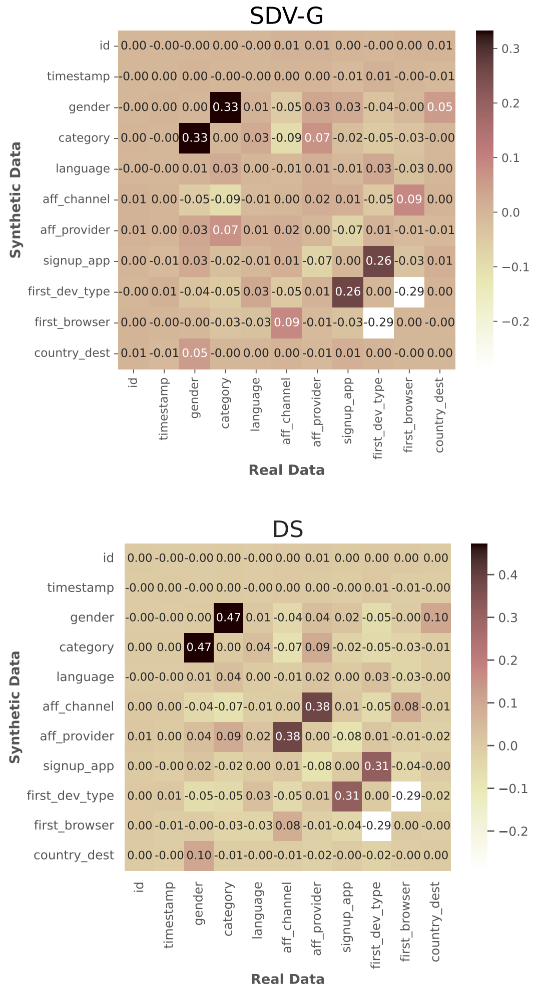
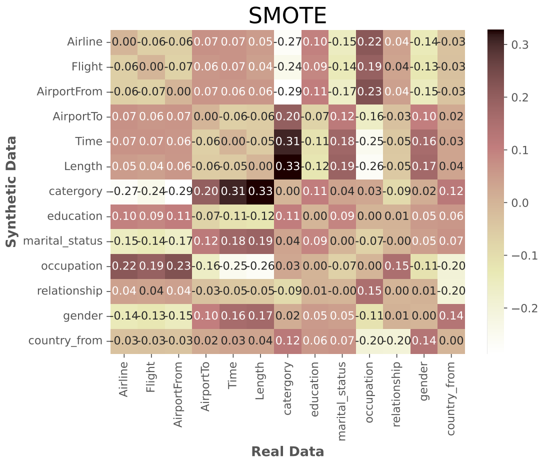
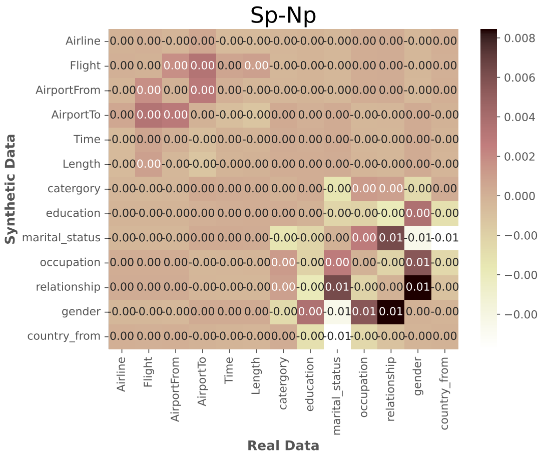
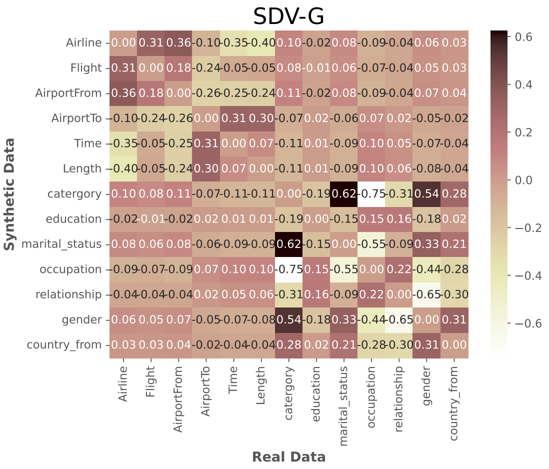


**Figure 3: Heatmaps of Airbnb Data**







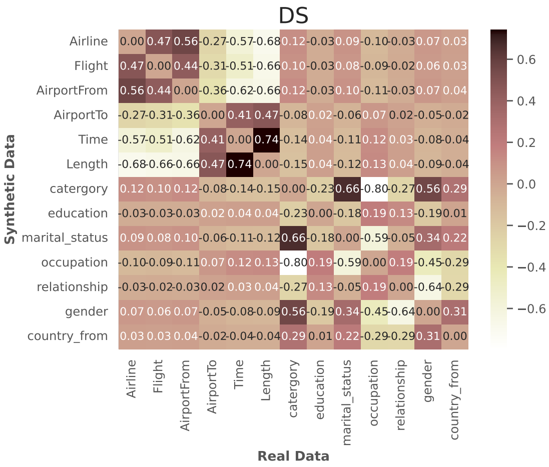


**Figure 4: Heatmaps of Airlines Data**

# Experimental Evaluation of Low Code development, Java Swing and JavaScript programming

André Calçada
Department of Systems and Computer Engineering,
Polytechnic of Coimbra, Coimbra Institute of Engineering
(ISEC), Rua Pedro Nunes, 3030-199 Coimbra,
+351239790200
a21250156@isec.pt

Jorge Bernardino
Department of Systems and Computer Engineering,
Polytechnic of Coimbra, Coimbra Institute of Engineering
(ISEC), Rua Pedro Nunes, 3030-199 Coimbra,
+351239790200, Centre for Informatics and Systems of the
University of Coimbra (CISUC), Polo II, Pinhal de
Marrocos, 3030-290 Coimbra, +351239790000
jorge@isec.pt

## ABSTRACT

Low Code is a technology that has been gaining popularity over the years, due to its potential and simplicity. But so far there has not been an experimental evaluation with other programming methods. This paper aims to introduce Low Code development technology and compare it with Java Swing programming and manual development with HTML (HyperText Markup Language), CSS (Cascading Style Sheet), and JavaScript. These technologies are compared using the following metrics: development time, execution time, and the number of written code lines. In this evaluation, two applications are implemented, a simple calculator, and a text editor, developed in all technologies. It is concluded that it is faster to develop applications in Low Code but in terms of execution time, these are usually slower. Although the Low Code development is still at a somewhat embryonic stage which leads to some bugs and errors, Low Code development is better in general than Java Swing programming, and somewhat similar to manual programming with HTML, CSS, and JavaScript. Another benefit is that Low Code generates HTML and CSS automatically.

## CCS CONCEPTS

• **General and reference** → Cross-computing tools and techniques; Evaluation; Cross-computing tools and techniques; Experimentation; • **Software and its engineering** → Software notations and tools; General programming languages; Language features.

## KEYWORDS

Java Swing, Low Code, JavaScript

**ACM Reference Format:**
André Calçada and Jorge Bernardino. 2022. Experimental Evaluation of Low Code development, Java Swing and JavaScript programming. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3548785.3548792

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548792

## 1 INTRODUCTION

Low Code development is a technology that facilitates programming by diminishing the handwritten code and allowing non-programmers to build and have a more active presence in the development process of the application. Low-code is a collection of tools that enables developers to avoid hand-coding and reduces the development effort of having an application ready for production.

There are many ways of developing an application but when it comes to programming, users sometimes have difficulties in choosing the language and the type of programming. For Low Code development, Neptune Planet 9 is used, a Low Code Development Platform (LCDP), for Java Swing programming NetBeans IDE (Integrated Development Environment), and for manual development with HTML, CSS, and JavaScript Visual Studio Code is applied.

Java is one of the most recognized and used programming languages, and Swing is a Java toolkit that allows programmers to easily build a User Interface (UI) for their Java applications. JavaScript is a lightweight, interpreted, object-oriented programming language mostly used to program web pages.

These technologies are chosen, Java Swing and JavaScript to compare with Low Code because Java Swing has a similar way of making the UI but runs on the operating system, unlike Low Code applications that run on a browser. JavaScript is chosen because it is the same type of programming that LCDPs use, but the UI development is manual.

These technologies can use database systems as external services, where the communication with the database is made through an API (Application Programming Interface) or another communication protocol, where a request is sent to the database and an answer is given in response. Depending on the database it is possible to manage the database through these requests, facilitating the development work.

In this paper, Low Code development, Java Swing programming, and manual development with HTML, CSS, and JavaScript are compared, by developing two applications in all types of programming, which has not been done before according to [1]. The platforms chosen to develop the apps are based on our experience with these types of platforms and because they are free of charge. The objective of this study is to help users to know what type of programming to choose through the comparative evaluation of the developed applications.

The rest of this paper is structured as follows. Section 2 introduces background concepts and the methodology applied. Section 3

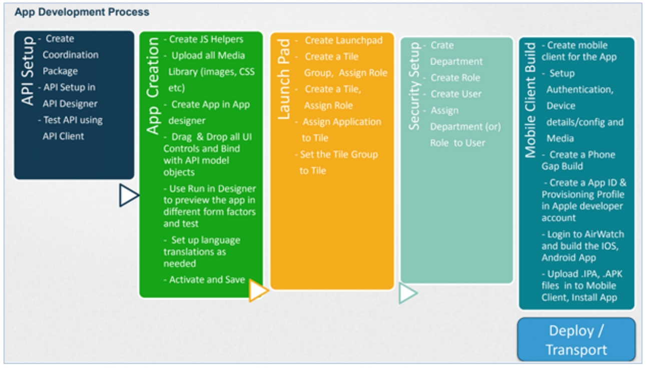

App Development Process
API Setup
- Create Coordination Package
- API Setup in API Designer
- Test API using API Client
App Creation
- Create JS Helpers
- Upload all Media Library (images, CSS etc)
- Create App in App designer
- Drag & Drop all UI Controls and Bind with API model objects
- Use Run in Designer to preview the app in different form factors and test
- Set up language translations as needed
- Activate and Save
Launch Pad
- Create Launchpad
- Create a Tile Group, Assign Role
- Create a Tile, Assign Role
- Assign Application to Tile
- Set the Tile Group to Tile
Security Setup
- Crate Department
- Create Role
- Create User
- Assign Department (or) Role to User
Mobile Client Build
- Create mobile client for the App
- Setup Authentication, Device details/config and Media
- Create a Phone Gap Build
- Create a App ID & Provisioning Profile in Apple developer account
- Login to AirWatch and build the IOS, Android App
- Upload .IPA, .APK files in to Mobile Client, Install App
Deploy / Transport

**Figure 1: Low code app development process. Source: [6].**

exhibits the results. Section 4 presents the discussion of the results. Section 5 describes the open problems. Finally, section 6 presents the main conclusions and future work.

## 2 BACKGROUND

This section introduces Low Code development, LCDPs, Java Swing, HTML, CSS, JavaScript, Node.js, IDEs, and the methodology used in the experiments.

### 2.1 What is Low Code?

The term ‘Low Code’ was introduced by Forrester market research company in June 2014, where these platforms are described as extraordinarily disruptive [2].

Low Code applications are developed using model-driven engineering principles and taking advantage of cloud infrastructures, automatic code generation, declarative, high level, and graphical abstractions to develop entirely functioning applications, meaning that these applications are mostly made through drag and drop of objects [3].

### 2.2 Low Code Development Platform

LCDPs emerged in the early 2000s helping development teams work faster. The Low Code Development Platform market started back in 2011, speeding up the development and maintenance processes.

An LCDP is set on a cloud or locally [3], allowing the development of Low Code applications using minimal code writing. Its objective is to give, to different types of users, the possibility to create applications in an easy, simple, and fast way [4].

Each LCDP has its programming language, such as Java, JavaScript, Python, and others [2]. LCDPs allow the development of distinct types of applications such as web apps and mobile apps [5].

The development of a Low Code application has the following processes: API Setup, App Creation, LaunchPad, Security Setup, Mobile Client Build, and Deploy/Transport, as shown in Figure 1.

API setup is a process where the developer creates APIs, which can be made in an LCDP or imported. The App Creation process is where the developer creates the app/module. In this process, LaunchPad is used to transform a module or various modules into a suitable application. The Security setup process is where is set up who has access to the application or which parts of it. Mobile Client Build process is where the application is prepared to become a mobile app, and at last Deploy. It should be mentioned that not all of these processes are required to implement an application, which depends on the type of application that is developed [6].

### 2.3 Java Swing

Java is an object-oriented language, and Swing is a widget toolkit GUI (Graphic User Interface). Swing started to be developed in 1996, supports multithreading, and allows Java programmers to easily create a UI for an application.

Some of the Swing components are labels (text), text areas, buttons, tables, frames (application page), combo box, scroll pane (object to scroll page), file chooser, (to get a file for reading or to save), menus, toolbars, and others [7].

### 2.4 HTML

HTML is the standard markup language for documents displayed in a web browser, defining their meaning and structure. There are also other technologies used to describe a web page’s appearance (CSS) or behaviour (JavaScript). “Hypertext” refers to links that allow users to create, store, and view text, connecting web pages directly so that “travelling” from one to another is quicker. Links are a fundamental characteristic of the Web.

HTML uses “markup” to distinguish text, images, and other content for display in a Web browser. An HTML element is detached from other text in a document by “tags”, which consist of the element name surrounded by “<” and “>”. HTML markup includes special “elements” such as <head>, <title>, <p>, <button>, and many others. The name of an element inside a tag is not case-sensitive, that is, it can be written in uppercase, lowercase, or a combination. For example, the <title> tag can be written as <Title>, <tItle>, or in any other style [8].

### 2.5 Cascading Style Sheet (CSS)

CSS is a stylesheet language for describing the presentation of elements in a HTML or XML (eXtensible Markup Language) document, including XML dialects such as SVG (Scalable Vector Graphics), MathML (Mathematical Markup Language) or XHTML (eXtensible HyperText Markup Language). CSS describes colours, layout, and fonts of Web pages, allowing to adapt the presentation to different types of devices. CSS is independent of HTML and can be used with any XML-based markup language.

CSS is one of the core languages of the web and is standardized across Web browsers according to W3C specifications [9]. Formerly, the development of various parts of CSS specification was synchronous, which allowed the versioning of the latest recommendations. There are new versions of CSS such as CSS1, CSS2.1, and CSS3. However, CSS4 has never been released.

Since CSS3, the scope of the specification increased considerably with CSS modules differing significantly. Therefore, it became more efficient to develop and release recommendations separately per module. Currently W3C, as an alternative of versioning the CSS specification, takes periodically a snapshot of the latest stable state of the CSS specification [10], [11].

## 2.6 JavaScript

JavaScript is a dynamic and lightweight scripting language, and it has broad participation in website and web application services. JavaScript has become one of the most widely used languages for Web development, however, many non-browser environments also use it, such as Node.js, Apache CouchDB, and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, which is used in web pages interface design, creating cookies, mobile apps, games, and so on.

JavaScript should not be confused with the Java programming language. The two programming languages have very different syntax, semantics, and uses [12]. The main difference between JavaScript and Java is that JavaScript code is written completely in text and needs only to be interpreted, while Java, on the other hand, must be compiled.

## 2.7 Node.js

Node.js is an increasingly popular event-driven architecture, open-source, cross-platform, back-end JavaScript runtime environment, widely used in server-side and desktop applications. Node.js executes JavaScript code outside a web browser and provides an effective asynchronous programming model. In Node.js, time-consuming IO operations, e.g., file access operations, can be delegated as asynchronous tasks, running in the dedicated threads. Thus, Node.js applications are not blocked by these time-consuming IO operations.

Node.js provides an effective asynchronous event-driven programming model and supports asynchronous tasks allowing developers to use JavaScript to write command-line tools and produce dynamic web page content before the page is sent to the user's browser. Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server-side and client-side scripts. These design choices aim to optimize throughput and scalability in web applications with many input/output operations [13].

## 2.8 Integrated Development Environment (IDE)

IDEs provide a convenient standalone solution that supports developers during various phases of software development and are designed to include all programming tasks in one application. One of the main benefits of an IDE is that they offer a central interface with the tools that a developer needs, including the following [14]:

- Code editor: Designed for writing and editing source code, these editors are distinguished from text editors because their function is to simplify and enhance the process of writing and editing code for developers.
- Compiler: Compilers transform source code that is written in a human-readable language into a machine-readable language.
- Debugger: Debuggers are used during tests and can help developers debug their application code.
- Build automation tools: These tools help developers to automate common developer tasks to save time.

Additionally, some IDEs may also contain [14]:

- Class browser: Used to study and reference properties of an object-oriented class hierarchy.
- Object browser: Used to inspect objects instantiated in a running application program.
- Class hierarchy diagram: This allows developers to visualize the structure of object-oriented programming code.

An IDE can be a stand-alone application, despite the fact it could be also included as part of one or more compatible applications [14].

## 2.9 Methodology

To compare Low Code development, Java Swing, and JavaScript programming, two applications are developed in all technologies. These applications are developed and tested on a computer with an Intel i5-8250U CPU, Windows 10, 512 GB SSD, and 8 GB of RAM (Random Access Memory).

The first application is a calculator with the basic math operations, sum, subtraction, multiplication, and division. The second is a simple text editor, like a simple notepad. In the implementations of these applications, the following metrics are assessed: development time, in minutes; execution runtime, in milliseconds (ms), this is the time that the application takes to set up the UI; the number of written code lines; and the operations execution time, in milliseconds. The execution runtime and operations execution time that is considered is the average of five executions of the application. To ensure that the results are viable, in JavaScript and Low Code, since an LCDP and its applications run on a browser, Brave browser, introduced in subsection 2.9.3 is used in incognito mode to get the execution time the browser cache is cleared before each run. Also, in Java Swing, before each run, CachemanXP program is executed to clean the RAM.

The calculator has a text area to show the input and results, and alongside the basic operation buttons, mentioned above, it has the numbers, from 0 to 9, the equal, the delete, the decimal point ".", and the clear buttons. The calculator doesn't give any errors, when inserting two operations it must do the first operation and with its result do the second (when introduced 5+2*10, the first operation is computed, 5+2=7, and then the second operation, 7*10=70), this may lead to mathematical miscalculation but it's not important for the evaluation. When there is an operation character in the text area, and another is inserted the first one must be replaced (if there is have "2+" in the text area and a "-" is entered, the result is "2-"). To get a valid operation, a number must be entered followed by an operation and another number and then press the equal button to get the result or add another operation.

The text editor is a simple text area with two buttons, one to load and the other to save the text. The text editor must give the user the possibility to choose where to save or load the file. The load and save functionalities must only allow .txt files.

The Low Code applications are implemented using Neptune Planet 9 LCDP, version 2.3.1, which will be introduced in subsection 2.9.1. In the Java Swing applications, NetBeans IDE is used, version 12.5, which is described in subsection 2.9.2.

The JavaScript applications are implemented with Visual Studio Code, version 1.63.2, which is presented in subsection 2.9.4. It should be noted that in pure JavaScript, it is not possible to let the user choose where to save a file, because it is always saved in the Downloads folder [15]. These platforms are chosen considering our knowledge about these types of platforms. NetBeans is chosen because it is one of the most user-friendly IDE, however, it does not support JavaScript. Due to this fact, with JavaScript, Visual Studio Code is chosen, which is one of the best IDEs, based on their usage and popularity [16].

The description of tests execution is presented in Tables 1 and 2 for the calculator and text editor, respectively. The tests are based on the following operations:

- "Add number" operation includes adding a number and the "." to the text area.
- "Add operation" adds a mathematical operator like "+" to the text area.
- "Add operation (S)" is the same as "Add operation". However, when there is already an operation character this is substituted, for example, when there is "1+" in the text area and a "-" is introduced, the result is "1-".
- "Add operation (R)" is the same as "Add operation". However, when there is already a valid equation its result is calculated. For example, "1+1+" turns "2+".
- "Delete" operation deletes a character of the text area.
- "Delete (N)" operation is when the text area is empty and the delete button is pressed.
- "Clear" operation clears the whole text area.
- "Result" operation calculates the result of the equation in the text area.
- "Result (N)" is the same as the result operation but when the equation is not valid, for example when there is a "1+" in the text area the equation is not altered, the value "1+", in the text area stays unchanged.
- "Load" operation loads the text from a .txt file to the text area
- "Save" operation saves the text in the text area into an existing .txt file. This is not possible to evaluate on JavaScript text editor, because when trying to save on an existing file, the browser creates a new file by adding "(1)" to the new file name.
- "Save (N)" operation saves the text in the text area into a new .txt file.

Some observations: 1) The calculator tests start with the calculator clear of values. 2) In the calculator tests the time to select a button is not considered, only the time of the operation. 3) The text editor tests do not take into consideration the time to write a text, and the same text is used for all tests. 4) In the text editor tests the time to select a file is not considered, only the time to save/load a file.

#### 2.9.1 *Neptune Planet 9.*

Neptune Planet 9 is an LCDP that uses as core technologies HTML, and CSS, and uses Node.js as the programming language. Its architecture is shown in Figure 2 and Figure 3.

The development of applications is done in the App Designer component, where data and resources from the Store, ODATA, Media Library, API Designer, and Server Scripts are used. Table Definition is used to define data types, which is not always necessary, as these types can be automatically imported from an external database. The LaunchPad serves as a Modules portal, where each developed module is added, which is possible with the use of Tiles that allows navigation for each module. Tiles are organized into groups through Tiles Groups. Users are configured in the Users component and can be created in the LCDP or obtained externally. In the LaunchPad each user has access to the modules depending on their role. Finally, the LaunchPad can run on the Web or in an APK (Android Package) that can be generated in the Mobile Client component for mobile devices [6].

Modules can be created from a workflow using the Workflow Designer component and the Theme Designer component can be used to create a predefined theme for the entire LaunchPad to visually enhance it [6].

#### 2.9.2 *NetBeans IDE..*

NetBeans is a free IDE where a programmer can develop applications in languages like Java, C, C++, PHP, and others. This IDE supports many platforms such as Windows, Linux, Solaris, and macOS, and it supports many types of API services. NetBeans like many other IDEs allows a programmer to develop many kinds of applications, from a plain text editor to a complex web app [17], [18].

#### 2.9.3 *Brave Browser.*

Brave is a free and open-source web browser developed by Brave Software, Inc. based on the Chromium web browser. Brave's popularity is on the increase, driven by privacy-by-default functionality, which automatically blocks online advertisements and website trackers. Brave is developed upon the open-source Chromium browser project which promotes faster and safer browsing. As the project is open source, Brave can make use of the code for their product, adding additional features on top. Privacy features include ad-blocking, antitracking functionality, and cryptocurrency offerings [19].

#### 2.9.4 *Visual Studio Code.*

Visual Studio Code is a cross-platform editor implemented by Microsoft for Windows, Linux, and macOS. In 2016, Visual Studio Code has progressed from the public preview stage and was released to the Web. Then, it has quickly become one of the top editors in terms of the popularity.

Visual Studio Code is a very powerful code-focused development environment expressly designed to make it easier to write web, mobile, and cloud applications using languages that are available to different development platforms and to support the application development lifecycle with a built-in debugger and integrated support for the popular Git version control engine [20].

## 3 EXPERIMENTAL RESULTS

This section presents the experimental results of the developed applications.

Figures 4, 5 and 6 presents the user interface of the calculator using Java Swing, Low Code, and JavaScript, respectively.

Figures 7, 8 and 9, presents the user interface of the text editor using Java Swing, Low Code, and JavaScript, respectively.

Figure 10 presents the text editor example of the *fileChooser* window for load operation.

Table 3 presents the results of the development of each application concerning development time, execution runtime, and handwritten code lines. In Tables 3, 4, and 5, Java Swing is referred to as JSW, Low Code as LC, and JavaScript as JSC.

Table 1: Calculator test execution

| Operation | Description |
|---|---|
| Add number | 1. Click on a number (random)<br>2. Read time of operation (1) |
| Add operation | 1. Click on an operation (random)<br>2. Read time of operation (1) |
| Add operation (S) | 1. Click on an operation (random)<br>2 Click on another operation (random)<br>2. Read time of operation (2) |
| Add operation (R) | 1. Click on a number (random)<br>2. Click on an operation (random)<br>3. Click on a number (random)<br>4. Click on an operation (random)<br>5. Read time of operation (4) |
| Delete | 1. Click on a number/s or operation/s (random)<br>2. Click on delete<br>3. Read time of operation (2) |
| Delete (N) | 1. Click on delete<br>2. Read time of operation (1) |
| Clear | 1. Click on a number/s or operation/s (random)<br>2. Click on clear<br>3. Read time of operation (2) |
| Result | 1. Click on a number (random)<br>2. Click on an operation (random)<br>3. Click on a number (random)<br>4. Click on equal<br>5. Read time of operation (4) |
| Result (N) | 1. Click on a number/s or operation/s (random)<br>2. Click on equal<br>3. Read time of operation (2) |

Table 2: Text editor test execution

| Operation | Description |
|---|---|
| Load | 1. Click on Load<br>2. Select a file<br>3. Read time of operation (2) |
| Save | 1. Write the text<br>2. Click on save<br>3. Select an existing file<br>4. Read time of operation (3) |
| Save (N) | 1. Write the text<br>2. Click on save<br>3. Choose file and name location<br>4. Read time of operation (3) |

Table 4 presents the average runtime of each operation of the calculator, in milliseconds.

Table 5 presents the runtime for each operation of the text editor in milliseconds.

## 4 DISCUSSION OF THE EXPERIMENTAL RESULTS

This section presents the discussion of the results of the previous section. To discuss these results it must be considered that Low Code and manual programming with HTML, CSS, and JavaScript are similar except Low Code generates HTML and CSS automatically.

### 4.1 Discussion of the Results: Comparing Low Code with Java Swing and JavaScript

As shown in Table 3 Java Swing and JavaScript applications have a better performance but at a bigger cost in terms of development time and hand-written code, compared to Low Code applications. Nevertheless, in a deeper analysis:

- In terms of development time, developing in Low Code is in average, 1.74 times faster than programming in Java Swing and 1.10 times faster than programming in JavaScript:
  - ○ In the Low Code calculator, development is 1.73 times faster than Java Swing.
  - ○ In the Low Code text editor, the development is 1.75 times faster than Java Swing.
  - ○ In the Low Code calculator, development is 1.06 times faster than JavaScript.

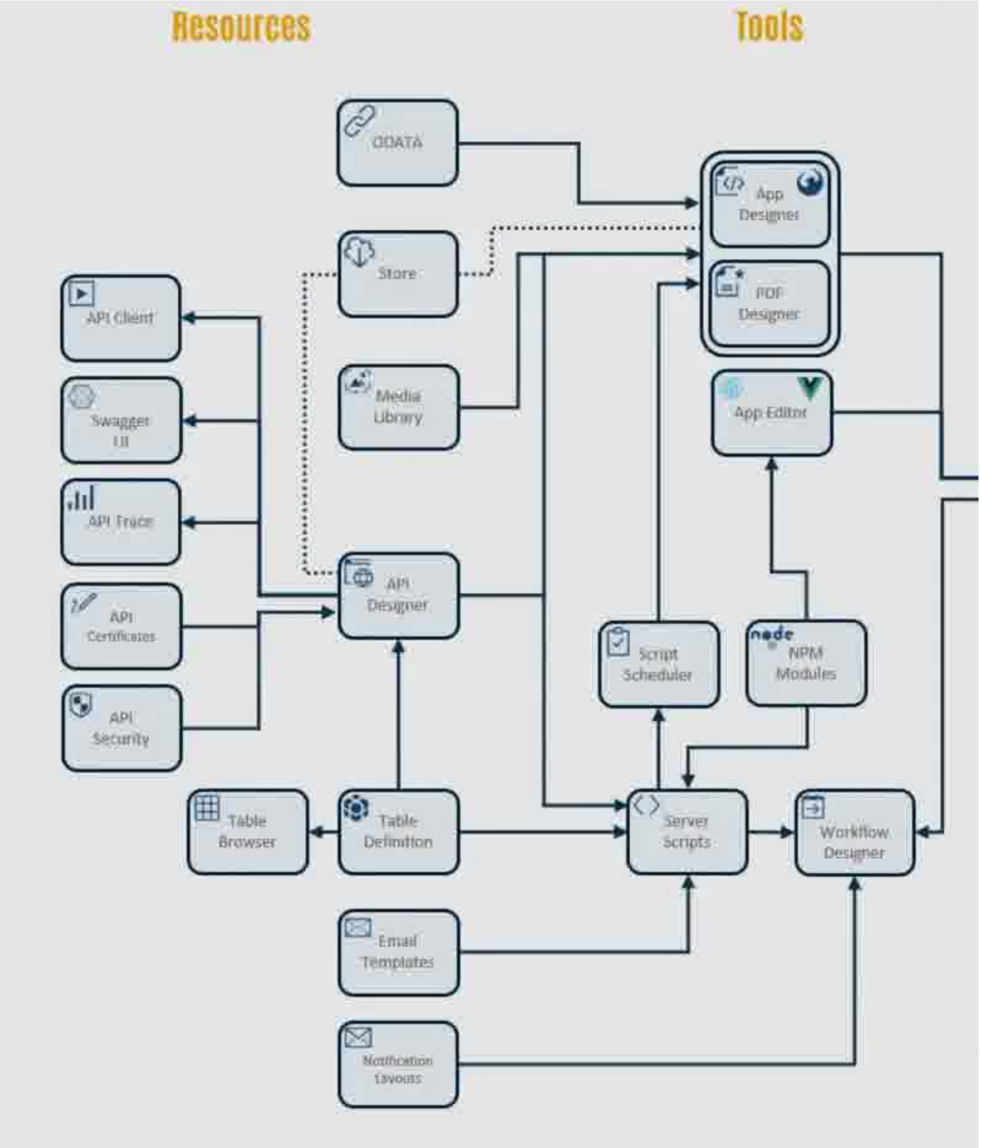


Figure 2: Neptune Planet 9 architecture, resources and tools. Source: [6].

Table 3: Results of the experiments

| Application / Property | Development time (minutes) | Execution runtime (ms) | Hand-written code lines |
|---|---|---|---|
| Calculator (JSW) | 57 | 156.34 | 72 |
| Calculator (LC) | 33 | 1082.60 | 64 |
| Calculator (JSC) | 35 | 37.40 | 106 |
| Text Editor (JSW) | 14 | 790.63 | 39 |
| Text Editor (LC) | 8 | 1494.60 | 12 |
| Text Editor (JSC) | 10 | 33.00 | 27 |

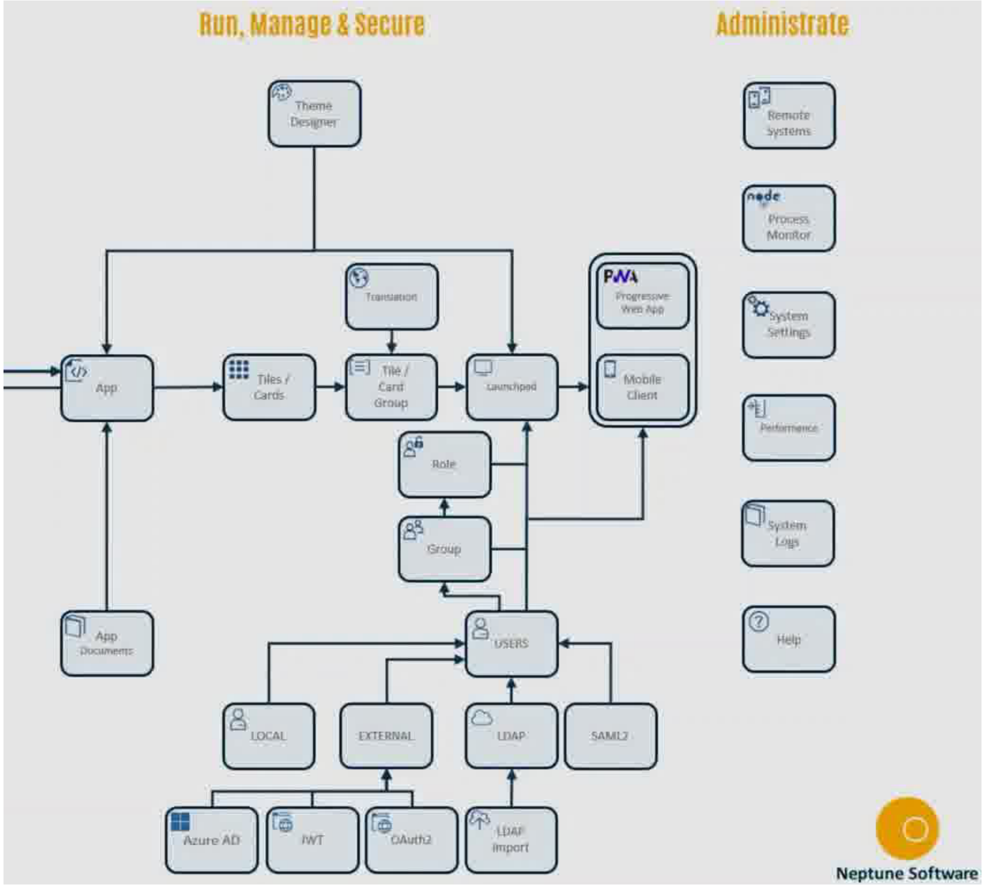


Figure 3: Neptune Planet 9 architecture, run, manage & secure and administrate. Source: [6].

Table 4: Results of the calculator operations in ms using JSW, LC, and JSC

| Operation | Calculator (JSW) | Calculator (LC) | Calculator (JSC) |
|---|---|---|---|
| Add number | 3.14 | 0.56 | 0.14 |
| Add operation | 3.25 | 0.52 | 0.16 |
| Add operation (S) | 1.05 | 0.74 | 0.06 |
| Add operation (R) | 1.78 | 0.80 | 0.48 |
| Delete | 0.42 | 0.26 | 0.10 |
| Delete (N) | 0.10 | 0.44 | 0.12 |
| Clear | 0.10 | 0.38 | 0.22 |
| Result | 1.26 | 0.22 | 0.60 |
| Result (N) | 0.21 | 0.44 | 0.16 |

Table 5: Runtime results for the text editor (in ms) using JSW, LC, and JSC

| Operation | Text editor (JSW) | Text editor (LC) | Text editor (JSC) |
|---|---|---|---|
| Load | 37.33 | 0.30 | 0.34 |
| Save | 5.66 | 0.92 | Not possible |
| Save (N) | 7.13 | 0.86 | 1.02 |

  - ○ In the Low Code text editor, the development is 1.25 times faster than JavaScript.
- In terms of execution runtime, Java Swing applications are in average, 2.72 times faster and JavaScript applications 36.61 times faster, when compared to Low Code applications:
  - ○ In Java Swing, the calculator, runtime is 6.92 times faster than Low Code.
  - ○ In Java Swing, the text editor runtime is 1.89 times faster than Low Code.

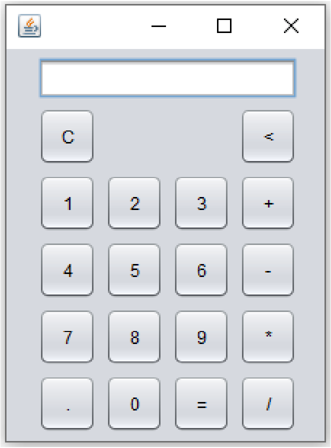


Figure 4: Java Swing calculator based on NetBeans.

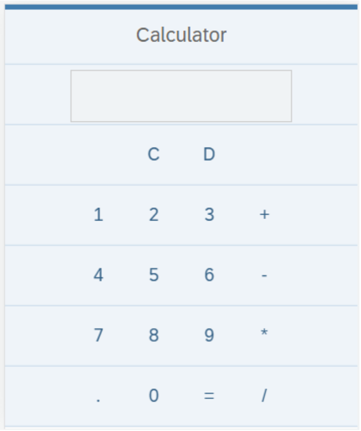


Figure 5: Low Code calculator based on Neptune P9 and Brave browser.

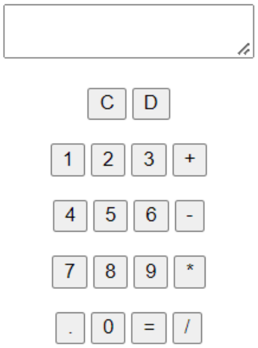


Figure 6: JavaScript calculator based on Visual Studio Code and Brave browser.

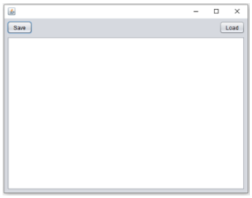


Figure 7: Java Swing text editor based on NetBeans.

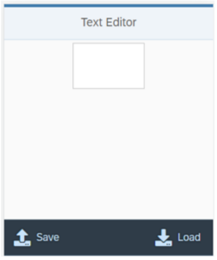
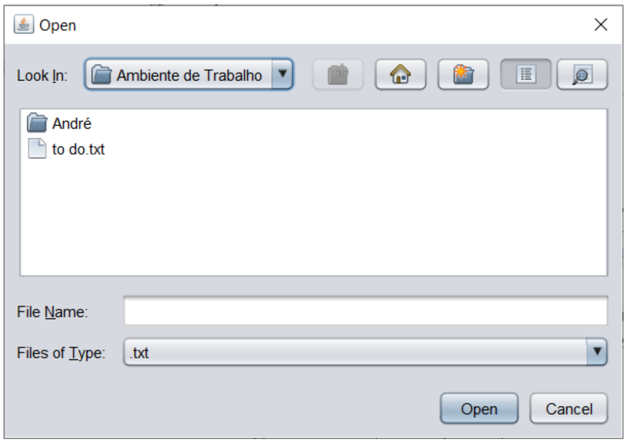


Figure 8: Low Code text editor using Neptune P9 and Brave browser.

Figure 9: JavaScript text editor using Visual Studio Code and Brave.



Look In: Ambiente de Trabalho

André

to do.txt

File Name:

Files of Type: .txt

Figure 10: Load operation *fileChooser* using NetBeans.

- In JavaScript, the calculator, runtime is 28.95 times faster than Low Code.
- In JavaScript, the text editor runtime is 45.29 times faster than Low Code.

- In terms of hand-written code lines, Low Code applications have in average, 2.18 times less code, compared to Java Swing and 1.75 times less code, compared to JavaScript:
  - ○ The low Code calculator has 1.13 times fewer code lines than Java Swing.
  - ○ The low Code text editor has 3.25 times fewer code lines than Java Swing.
  - ○ The low Code calculator has 1.66 times fewer code lines than JavaScript.
  - ○ The low Code text editor has 2.25 times fewer code lines than JavaScript.

As presented in Table 4 Low Code calculator has a better performance in terms of operation execution runtime, compared to the Java Swing calculator, and it is similar to the JavaScript calculator. Considering all the operations:

- In average the calculator operations are 2.14 times faster in JavaScript than in Low Code and 2.59 times faster in Low Code than in Java Swing.
- Java Swing calculator is faster in three operations "Delete (N)", "Clear" and "Result (N)". This is because Java is usually faster and, in this case, there is only one line of code to execute (in "Delete (N)" there's an if, in "Clear" there's a set value, in "Result (N) there's an if), and the difference, compared to Low Code, is 0.34, 0.28, and 0.23, respectively. In each operation there exists a negligible difference, thus considering that it is much slower in other operations.

As shown in Table 5, the Low Code text editor has a better performance in terms of operation execution runtime. For example:

- "Load" operation is 124.43 times faster in Low Code than in Java Swing, and 1.13(3) times faster than JavaScript.
- "Save" operation is 6.15 times faster in Low Code than in Java Swing.
- "Save (N)" operation is 8.20 times faster in Low Code than in Java Swing and 1.19 times faster than in JavaScript.

The Low Code text editor is in average of execution runtime of operations, 24.10 times faster than the Java Swing text editor, and 1.17 times faster than the JavaScript text editor. Low Code and JavaScript are faster than Java Swing in this case because unlike JavaScript, Java Swing needs to understand what operating system it is running on to make a system call, JavaScript does not need that because it is managed by the browser [21].

Comparing only Java Swing and JavaScript depends on the application's complexity and nature and it is presented in the next subsection.

### 4.2 Discussion of the Results: Java Swing with JavaScript

As presented in Table 3, JavaScript applications have better performance and take less time to develop than Java Swing but need more handwritten code lines. Making a deeper analysis:

- In terms of development time, developing in JavaScript is, in average, 1.58 times faster than programming in Java Swing.
  - ○ In the JavaScript calculator, development is 1.63 times faster.
  - ○ In the JavaScript text editor, the development is 1.40 times faster.
- In terms of execution runtime, JavaScript applications are, in average, 13.45 times faster than Java Swing applications.
  - ○ In JavaScript, the calculator, runtime is 4.18 times faster.
  - ○ In JavaScript, the text editor runtime is 33.96 times faster.
- In terms of hand-written code lines, Java Swing applications have, in average, 1.20 times less code, compared to JavaScript.
  - ○ In the Java Swing calculator, the number of code lines is 1.47 times less.
  - ○ In the Java Swing text editor, the number of code lines is 1.44 times less.

As shown in Table 4, the JavaScript calculator has a better performance in terms of operations execution runtime, compared to the Java Swing calculator. Considering all the calculator operations:

- In average the calculator operations in JavaScript are 5.54 times faster than in Java Swing.
- Java Swing calculator is only faster in two operations "Clear and "Delete (N)" and the difference is 0.02 and 0.12 milliseconds, respectively, which is a negligible difference, considering that Java Swing is much slower in other operations. JavaScript calculator is 5.54 times faster than Java Swing calculator in average of runtime execution of all operations.

As presented in Table 5 JavaScript text editor has a better performance in terms of operation execution runtime. Making a deeper analysis:

- "Load" operation is 109.79 times faster in JavaScript than in Java Swing.
- "Save" operation is not comparable because it cannot be tested in the JavaScript text editor.
- "Save (N)" operation is 6.99 times faster in JavaScript than in Java Swing.

The JavaScript text editor is 32.69 times faster than the Java Swing text editor considering the average execution runtime of all operations.

### 4.3 Summary of all Results

Table 6 presents the average of the operations, and Table 7 the standard deviation for all the previous results taking into consideration the number of code lines, execution runtime (ms), development time (minutes), and operations runtime (ms). These results are obtained from Table 3 and from Tables 4 and 5.

**Table 6: Operations average for all experiments**

| Operation | Java Swing | Low Code | JavaScript |
|---|---|---|---|
| Code lines | 55.50 | 38.00 | 66.50 |
| Execution runtime | 473.49 | 1288.60 | 35.20 |
| Development time | 34.00 | 20.50 | 22.50 |
| Operations runtime | 5.12 | 0.54 | 0.310 |

**Table 7: The standard deviation for all experiments**

| Operation | Java Swing | Low Code | JavaScript |
|---|---|---|---|
| Code lines | 16.50 | 26.00 | 39.50 |
| Execution runtime | 317.15 | 206.00 | 2.2 |
| Development time | 20.00 | 12.50 | 12.50 |
| Operations runtime | 9.95 | 0.23 | 0.28 |

With the results of Tables 6 and 7, the differences between these technologies can be seen more clearly:

- Low Code is 9.48 times faster than Java Swing and 1.74 times slower than JavaScript, in terms of operation runtime.
- JavaScript is 16.52 times faster than Java Swing, in terms of operation runtime.
- JavaScript has the lowest standard deviation compared to the other technologies, except in the number of code lines, where Java Swing has the lowest standard deviation.

Note that the values comparing these technologies may vary due to the application's complexity and its nature.

## 5 OPEN PROBLEMS

This section presents the problems addressed in this paper. The comparison of Low Code, Java Swing, and Java Script technologies can be addressed in many ways. In this paper, each technology was compared by looking at some of its application's metrics, like execution runtime. This practical approach gives developers a perspective of how these technologies can help them in their job.

As described in [1], there is no comparison of Low Code with other technologies, which leaves space for this type of research. This gap was filled with our research, but there is still significant work to be done. We identify the following open research problems:

- Comparison with other Low Code technologies.
- Using Low Code with more complex applications including database development.
- Comparing the frontend UI, backend logic, and data store, to be developed using Low Code technologies.
- Using machine learning in Low Code vs machine learning in other technologies.
- Research of Low Code Development Platforms usage for communication, human behaviour, and decision-making.

## 6 CONCLUSIONS AND FUTURE WORK

Low Code development is a technology that speeds up the process of deploying an application version to the production environment. Low Code facilitates programming by diminishing the handwritten code and allowing non-programmers to build applications. These technologies may also simplify the work of database developers.

The Low Code development, Java Swing, and JavaScript programming have been experimentally evaluated and it can be concluded that Low Code applications are valuable when what is important is the development time and writing code. However, their runtime execution for the setup of the application is slower than Java Swing and JavaScript. Low Code and JavaScript are faster at executing most of the operations than Java Swing which leads to the conclusion that Low Code and JavaScript applications have a better performance. Their performance in terms of operation execution time is very similar, which occurs because Node.js is based on JavaScript and the applications of these technologies run on the same environment, a browser.

Despite the advantages of Low Code, it must be taken into consideration that Low Code has a big learning curve. It should also be noted that the compared technologies run in different environments, Java Swing runs on the operating system, and the other technologies run on a browser, which directly impacts performance.

It can be concluded that JavaScript is the best programming method in terms of execution runtime, although it may be the one with a larger number of code lines, depending on the applications.

As future work is intended to develop a study with other technologies, like .Net, and with more complex applications, such as a web app that manages users, so there can have a thorough comparison.

## REFERENCES

[1] N. Prinz, C. Rentrop, and M. Huber, "Low-Code Development Platforms – A Literature Review," AMCIS 2021 Proceedings, 2021.
[2] Y. Luo, P. Liang, C. Wang, M. Shahin, and J. Zhan, "Characteristics and Challenges of Low-Code Development," 2021, doi: 10.1145/3475716.3475782.
[3] A. Sahay, A. Indamutsa, D. Di Ruscio, and A. Pierantonio, "Supporting the understanding and comparison of low-code development platforms," Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020, 2020, doi: 10.1109/SEAA51224.2020.00036.
[4] C. Silva, J. Vieira, J. C. Campos, R. Couto, and A. N. Ribeiro, "Development and Validation of a Descriptive Cognitive Model for Predicting Usability Issues in a Low-Code Development Platform," Human Factors, vol. 63, no. 6, 2021, doi: 10.1177/0018720820920429.
[5] "No-code/low-code: why should you be paying attention." https://venturebeat.com/2021/02/14/no-code-low-code-why-you-should-be-paying-attention/ (accessed Nov. 04, 2021).
[6] "Neptune." https://www.neptune-software.com/ (accessed Feb. 12, 2020).
[7] B. Cole, R. Eckstein, J. Elliott, M. Loy, D. Wood, and O. ' Reilly, "JavaTM Swing, 2nd Edition," 2002.
[8] "HTML: HyperText Markup Language | MDN." https://developer.mozilla.org/en-US/docs/Web/HTML (accessed Mar. 11, 2022).
[9] "Cascading Style Sheets." https://www.w3.org/Style/CSS/#specs (accessed Jan. 18, 2022).
[10] "CSS: Cascading Style Sheets | MDN." https://developer.mozilla.org/en-US/docs/Web/CSS (accessed Jan. 18, 2022).
[11] H. Wium. Lie and Bert. Bos, "Cascading style sheets: designing for the Web," p. 396, 1999.
[12] "JavaScript | MDN." https://developer.mozilla.org/en-US/docs/Web/JavaScript (accessed Jan. 18, 2022).
[13] "Node.js." https://nodejs.org/en/ (accessed Jan. 31, 2022).
[14] "What is an IDE or Integrated Development Environment?" https://www.veracode.com/security/integrated-development-environment (accessed Nov. 30, 2021).
[15] "Mozilla | MDN - downloads." https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/downloads/download (accessed Jan. 24, 2022).
[16] "TOP IDE index." https://pypl.github.io/IDE.html (accessed Nov. 30, 2021).
[17] Tim Boudreau, Jesse Glick, Simeon Greene, Vaughn Spurlin, and Jack Woehr, NetBeans: The Definitive Guide: Developing, Debugging, and Deploying Java Code, 1st ed. O'REILLY, 2002.
[18] "Apache NetBeans." https://netbeans.apache.org// (accessed Nov. 18, 2021).
[19] "Secure, Fast & Private Web Browser with Adblocker | Brave Browser." https://brave.com/ (accessed Dec. 30, 2021).
[20] "Visual Studio Code - Code Editing. Redefined." https://code.visualstudio.com/ (accessed Jan. 13, 2022).
[21] "System Calls in Operating System - javatpoint." https://www.javatpoint.com/system-calls-in-operating-system (accessed Feb. 16, 2022).

# Applications of Majority Judgement for Winner Selection in Eurovision Song Contest

Areeba Umair
University of Naples Federico II
Naples, Italy
areeba.umair@unina.it

Elio Masciari
University of Naples Federico II, ICAR-CNR – Institute of Italian National Research Council
Naples, Italy
elio.masciari@unina.it

Giusi Madeo
IC Rende Commenda
Rende (CS), Italy
ing.giusimadeo@yahoo.it

Muhammad Habib Ullah
University of Naples Federico II
Naples, Italy
muhammad.habibullah@unina.it

## ABSTRACT

The existence of big data, social media interactions, and digital globalization has changed the way people make decisions either in their life or those of collective importance. Computational Social Choice (COMSOC), as an emerged field, has tried to join various social fields (social choice theory) and technical fields (computer science, mathematics, economics and logic). In the last few decades, expert rating was used to select the winner in the contest or competition, that was, later, merged with crowd voting. However, the results of voting based on aggregation of crowd opinion was not considered satisfied. The majority judgement is a new method of election. It is the consequence of a new theory of social choice where voters judge candidates instead of ranking them. In this research, we used Eurovision song contest data of 2021 final round. Eurovision song contest is held annually, in which almost 40 countries participate. We applied majority judgement on the Eurovision song contest data and found that Italy got highest position, followed by Croatia and Australia acquiring second and third positions respectively in competition of 2021.

## CCS CONCEPTS

• **Information systems** → *Data analytics*; • **Computing methodologies** → **Distributed artificial intelligence**; • **Computer systems organization** → **Robotic autonomy**; • **General and reference** → *Surveys and overviews*; • **Networks** → *Wireless local area networks*; • **Hardware** → *Sensors and actuators*; *Wireless devices.*

## KEYWORDS

Majority Judgement, Computational Social Choice, Eurovision Song Context

**ACM Reference Format:**
Areeba Umair, Elio Masciari, Giusi Madeo, and Muhammad Habib Ullah. 2022. Applications of Majority Judgement for Winner Selection in Eurovision Song Contest. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3548785.3548791

## 1 INTRODUCTION

There is a paradigm shift towards Computational social choice after the big data changes the way of data collection and processing. To make correct decision by using the information acquired from the big data is a challenge [17], [42]. Computational social choice, also known as COMSOC, has emerged as a interdisciplinary field during the last few decades and it acts as a bridge between the social science and technical science as well as between the classic and modern problems [3]. It draws knowledge from social choice theory, computer science, mathematics, economics and logic [31]. The social choice theory focuses on how the individual votes are aggregated in such as way that the society make the collective decisions [28] [33]. COMSOC introduces an algorithmic concept in the social choice theory where the main focus in on the computational and algorithmic perspectives of voting challenges [10]. The voting process is the central area of COMSOC. According to [17], the most known voting rules are:

- Majority: The candidate with majority votes is the winner
- Plurality: The candidate must have more votes as compared to other candidates
- Borda count method: The candidates with highest score is the winner
- K-approval: The candidate with more approvals is the winner
- Copeland's method: The candidate with the highest number of pairwise victories
- Veto: The candidate with the low negative score is the winner

The electing and ranking candidates has been a challenge since many decades [7], [15]. All around the world, different nations, societies, organizations, communities elect their representatives as well as juries or committees rank the employees, singers, drivers, artists, students, universities, hospitals, nominees for different awards, movies, beauty etc. [23] [27]. With the development of intelligent DSS (decision support system), the degree of uncertainty during election and ranking can be reduced [13] [16], [41]. In past

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08.
https://doi.org/10.1145/3548785.3548791

decades, the small group of people, known as experts, used to select the best one from the pool of candidates and this mechanism is called "expert rating". Recently, instead of relying on few experts, one can easily get the opinion of the community users publicly for the selection of winner. This approach is called crowd sourcing [12]. Traditionally, the experts are professional people who have specialized knowledge and exposure about their particular field. However, the crowd are a number of people from different backgrounds and exposures [25]. Today, with the help of social media and other platforms, it is easy for the crowd to share their opinion over the internet [14]. In some cases and fields, the crowds opinion is more excellent as compared to the experts [43].

Another ranking methods theory is "Condorcet's Paradox" developed by French philosopher Nicolas de Condorcet, in 18th century. According to Condorcet Jury theorem, if each member of a jury has an equal and independent chance better than random, but worse than perfect, of making a correct judgment on whether a defendant is guilty (or on some other factual proposition), the majority of jurors is more likely to be correct than each juror and the probability of a correct majority judgment approaches as the jury size increases [8]. It is observed that the majority preferences can be termed as irrational even though the individual preferences are rational [9], [24]. Kenneth Arrow developed another ranking theorem in 20th century [19]. He proved that there exists no method of aggregating the preferences of two or more individuals over three or more alternatives into collective preferences, where this method satisfies five seemingly plausible axioms: (1) universal domain; (2) ordering; (3) weak Pareto principle; (4) independence of irrelevant alternatives and (5) non-dictatorship.

Balinski and Laraki introduce new voting system called, Majority Voting (MJ) [21] The majority judgement is a new method of election where the jury or voters only judge the candidate instead of ranking them. It is a consequence of a new theory of social choice [6]. In majority judgement, the the jury of judges or voters do not vote themselves, but they evaluate the candidates in some common formats of grades. The majority judgement gives the solution of search for an optimal methods of competition or election provided that the merits of competitors are to be evaluated [4]. The majority judgement has been practices in competitions around the world as well as in political scenarios. The grades given by voters are considered as input while the majority grade for each candidate is produced as output. This majority grade is used to rank the order of candidates or to select a winner [44]. The majority grade is the new process, which ensures that if the majority of the voters gives grade B to any candidate, its majority vote will be B. However, if everyone of the majority give grade C to the candidate, that candidate majority grade will certainly not be C [35].

The Eurovision Song Contest (ESC) is an annual event and was started in 1955 and held in Lugano, Switzerland in 1956 for the first time with seven participants countries. Later, in 1961, the number of participant countries increased to 16. Now-a-days, the non-European countries can also participate in the context such as Israel, Morocco, and Turkey, which have become the regular participant now. Almost several hundred millions of people watch the contest [26]. In the Eurovision song contest, the participant countries give votes for the selection of winner. However, there exist the suspicions and accusations of 'tactical' and 'political' voting in the song contest [39]. Eurovision is the most widely viewed event in all the candidate countries. It is also covered by the press as being the most famous international contest. The winner country celebrate the victory nationally. Often the leader of the winner country congratulate to the whole team and the whole nation for the victory and highlights the importance of victory towards their country and national morale [1], [22]. In Eurovision song context, the voting system was modified several times. In starting, the national juries used to select the winners. Later, in 1997, televoting was introduced. Then, from 2009, a hybrid system was introduced containing the popular vote and a jury vote to select the winner. In 2016, it was again revolutionized where the votes of jury and televoting were arranged separately [40]. Nevertheless, we believe that applying majority judgement in Eurovision song context can result in more fair voting.

## 2 RELATED WORK

Many researchers worked in the field of complexity of query evaluation. In [2], the researchers computed possible and certain answers using partial order theory and tried to reduce the complexity in their particular topic. In [31], a systematic investigation query evaluation on election databases was carried out by the researchers. They analyzed the interaction between the partial preferences, the voting rules and the relational context impacts on the complexity of query evaluation. Moreover, the researchers studied the computational complexity of the evaluation problem for approval voting and positional scoring rules regarding PPIC, the Mallows noise model, and EDM [7]. The researchers in [11], worked on the complexity of the possible winners (PW) on partial chains considering partial order theory. The article [10], investigated the practical aspects of computing the necessary and possible winners in elections over incomplete voter preferences. Dixit et al. used SAT solving i their research and investigated the aggregation queries over inconsistent databases [16].

The challenges of computational social choice for voting in multi-agent systems have been described in [17] while the authors in [18] explained the advantages and disadvantages of the crowd voting and expert voting. Imber et al. studied the complexity of some of the computational problems for the classic approval-based committee voting rules [27]. In [28], the authors explored the evaluation of similarity of voting procedures. [29], the authors studied the complexity of estimating the probability of an outcome in an election over probabilistic votes. Kovacevic et al. used the machine learning for the fusion of crowd and export opinion in crowd voting environment [32] while [43] used the machine learning along with crowd voting for the stock model selection. Various authors work on the preference aggregation as well [36], [34]. A comparison of the studies related to COMSOC is given in Table 1.

**Table 1: State-of-the-art in the field of COMSOC**

| REF | Methodology | Application | Theory Type | Solution |
|---|---|---|---|---|
| [2] | Conceptual | Computing possible and certain answers | Partial order theory | Complexity |
| [7] | Conceptual | Predicting election outcomes and estimating their robustness | Mallows noise model | Complexity |
| [11] | Conceptual | Possible winners on partial chains | Partial order theory | Complexity |
| [10] | Experimental | Necessary and possible winners | Partial order theory | Intractability |
| [16] | Conceptual | Answering aggregation queries | N/A | N/A |
| [17] | Comparative | Voting in multi-agent systems | N/A | N/A |
| [18] | Conceptual | Votes from crowd and knowledge from experts | Collective decision | Decision |
| [27] | Conceptual | Committee voting | Collective decision | Complexity |
| [28] | Conceptual | Multi-agent systems and voting | Condorcet theory | Polynomial time |
| [29] | Conceptual | Outcomes in probabilistic elections | N/A | Probability |
| [30] | Conceptual | COMSOC meets databases | Partial order | Complexity |
| [31] | Conceptual | Query evaluation in election databases | N/A | Complexity |
| [32] | Experimental | Fusion of crowd and experts in crowd voting | Collective decision | N/A |
| [33] | Conceptual | Decision making under incomplete knowledge | Collective decision | Efficiency Stability |
| [38] | Conceptual | Online elicitation of necessarily optimal matchings | Partial order | Complexity |
| [19] | Conceptual | Multi-robot task allocation problem | Arrow's theorem | N/A |
| [12] | Conceptual | Crowd voting on participation in crowdsourcing contests | Expectancy theory and tournament theory | Contest's reliance |
| [15] | Conceptual | Frugal bribery in voting | N/A | Polynomial time solvable |
| [14] | Experimental | Online review sites affect collective decision making | Collective decision | N/A |
| [13] | Conceptual | Information and analytical collective decision-making | Collective decision | Efficiency |
| [34] |  | Computational social choice and preference reasoning | Preference reasoning | N/A |
| [36] | Experimental | Making group decisions | Collective decision | Intractability |
| [20] | Conceptual | Support decision making on university program selection | Collective decision | Leveraging decision making |
| [43] | Experimental | Stock selection model | Collective decision | Stock recommendations |

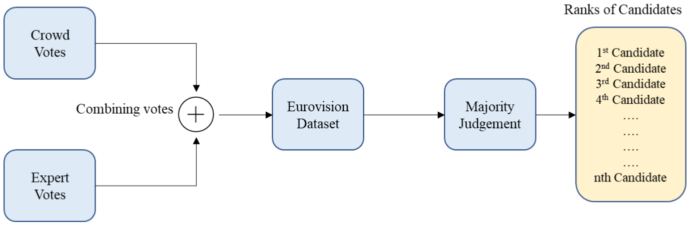


**Figure 1: Proposed Framework**

## 3 METHODOLOGY

The overall methodology used in this research us given in figure 1. We first of all get the jury votes and expert votes and combine them manually. A very careful approach was kept while adding the votes manually. Then, majority judgement was applied over the newly created dataset of Eurovision Context. The majority judgement provided the ranks of all the participating countries.

### 3.1 Dataset

In this research, we have used the Eurovision 2021 song competition dataset. The 65th edition of Eurovision song competition was held in 2021, in which 39 nations participated. Only 26 nations were selected for the final round. The final score was the aggregate of jury and televotes from all the 39 participants. The data in the rows shows the votes given by each member to the finalists. The method of decision was that top 12 songs were selected and to be ranked in such as way that the 12 points must be for the best song and the 1 point for the 12th choice, while other participants get 0 points. Self voting was not allowed in the competition.

### 3.2 Majority Judgement

In this research, we ranked the candidates of Eurovision song context using majority judgement method. The majority judgement method, a new method of voting, was proposed by Balinski and Laraki in 2007 [5]. It demonstrates that Majority Judgment is the only method that meets a whole set of criteria that have been developed over several centuries in the field of “social choice theory”. It avoids the famous paradox of Jean Antoine Caritat, Marquis de Condorcet, and its generalization to Kenneth Arrow's "impossibility" theorem, by considering the problem of voting differently. Instead of imagining that a voter has an ordered list of candidates in their head (which this experiment belies), it is assumed that they can rate each candidate directly.

The purpose of this voting procedure is to selection one out of n candidates $(n \geq 2)^2$. In song contexts, each judge or the voter awards a certain grade to the participants, that is measure on ordinal scale. These ordinal grades are expressed in numbers such as 1,2,3,4.....10 or in words such as excellent, good, fair, acceptable, bad, poor. The next step is to determine the median grade. Median grade is the median of the all the grades of the candidates. The candidate with the highest median grade is selected as a winner. In the case of tie breaking between the median grades, $\alpha$, [5] defined some tie breaking rules:

(1) Delete the $\alpha$ from the tied candidates.
(2) Compute the new value of the median grade, $\alpha$.
(3) If the new value of $\alpha$ of any previously tied candidate is higher than the others, that candidate is selected as winner. If there is still tie between the new values of $\alpha$, the process should be repeated from step one.

The majority value of each candidate is assigned in the form of ordered triple [6].

$$(p, \alpha^*, q) where: \begin{cases} \alpha = & \text{median grade of the candidate or majority grade} \\ p = & \text{number of grades above the majority grade} \\ q = & \text{number of grades below the majority grade} \end{cases}$$

The $^*$ can be 0, positive or negative depending upon the values of p and q.

$$\alpha^* = \begin{cases} \alpha^+, & \text{if } p > q \\ \alpha^0, & \text{if } p = q \\ \alpha^-, & \text{if } p < q \end{cases}$$

if there are two candidates suppose A and B, who have majority value of $(p_A, \alpha_A, q_A)$ and $(p_B, \alpha_B, q_B)$ respectively. A ranks higher than B and $(p_A, \alpha_A, q_A)$ ranks higher than $(p_B, \alpha_B, q_B)$, if

- A's majority grade is higher than B's (or $\alpha^*_A > \alpha^*_B$)
- Both of them have $\alpha^+$ majority grade and $p_A > p_B$
- Both of them have $\alpha^-$ majority grade and $q_A < q_B$

The winner is the applicant who has a maximal majority-value triple in this ordering. If there exists numerous such applicants, then probably the winner ought to be selected through lottery from amongst them.

## 4 EXPERIMENTS, RESULTS AND DISCUSSION

We have used the Ranky library of python to implement the majority judgement [37]. We consider the list of candidate countries C=(c1,c2,...c39) and a list of voter countries V=(v1,v2,...v26). A score matrix M of size n x m is obtained by scoring the performance of each finalists country by each voter country (C is the list of rows of M and V is the list of columns of M). All the candidates countries were scored using the same procedure i.e. majority judgement. The challenge is to obtain the single rank of each country using the r = rank(f(M)) from score matrix.

The results of the experiments are given in Table 2, where Italy stands first, Croatia stood second and Australia got third position.

| Country | Max. Rank | Position |
|---|---|---|
| Italy | 4.5 | 1st |
| Croatia | 3.5 | 2nd |
| Australia | 3 | 3rd |

**Table 2: Results of Eurovision song context using majority judgement**

Figure 2 shows the rank of each participating country. It is clear from the figure that the winner country stands at the highest point. Considering the results, and our dataset, we concluded that Italy is the country with highest rank, followed by Croatia and Australia with second and third position respectively. The figure 3 gives the visualization of preference matrix (2D).

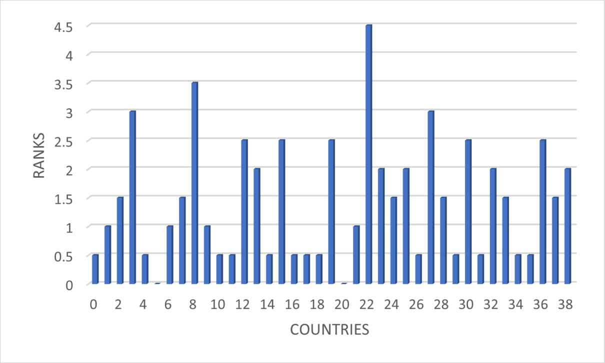


**Figure 2: Ranking of participants of Eurovision song context using majority judgement**

While discussing the results of voting, we should also consider the two important factors.

**1. Cultural and linguistic similarities and differences**: First is cultural and linguistic similarities and differences, which means that common musical taste of two or more than two nations results in the strong preference of voting for a particular country. There is possibility that two countries might have similar traditions or culture and they are familiar with each other's rituals very well. So, in such as case, there are greater chances of likeness between such countries. Moreover, language understanding in song context also matters alot. People will definitely like those songs which they fully understand and hence, can enjoy the lyrics. Therefore, the countries with same or similar languages can have a higher tendency to vote each other.

**2. Voting bias** Voting bias is also a relative phenomena, because geographical factors or issues strongly affect the behaviour of voter countries in the Eurovision song context. It was observed that many countries like or dislike the songs of their neighbouring countries. This can lead to the fact that geographical effects cause political voting in the contest [39].

## 4.1 Remarks of the MJ procedure in our case study

There are several advantages of using majority judgement in our case study:

- All voters were considered equally
- All candidates are considered equally
- If a candidate A is awarded highest grades from all the voters, then A is the winner
- All candidates are ranked in transitive order
- A winner A is still a winner if a candidate y is removed
- If the grades of the winner A is increased, s/he will again a winner.
- The winner A is still a winner, if a new candidates is added with the identical grade distribution that of A or other candidate.
- It tends to reduce manipulation in voting.
- Conceptually, the motivations of electorate and their satisfaction are modelled with the aid of their "utilities." The utility feature of a voter is complicated and absolutely unknown. It's miles probable to imagine that a voter would like a candidate's final grade to be as close as feasible to the grade he or she believes the candidate deserves, but it is not always so! Within the "plausible" case, the candidate's utility function is absolute, on the other hand it will become relative, i.e., what matters are the candidates' very last rankings now not their very last grades. However anyways, the majority judgement mechanism makes no assumptions anything about the voters' utilities. It depends simplest on what may be recognised in practice. It is far "strategy-proof" for huge training of affordable software functions, and, when not anything is understood approximately them, it excellent combats strategic manipulation.

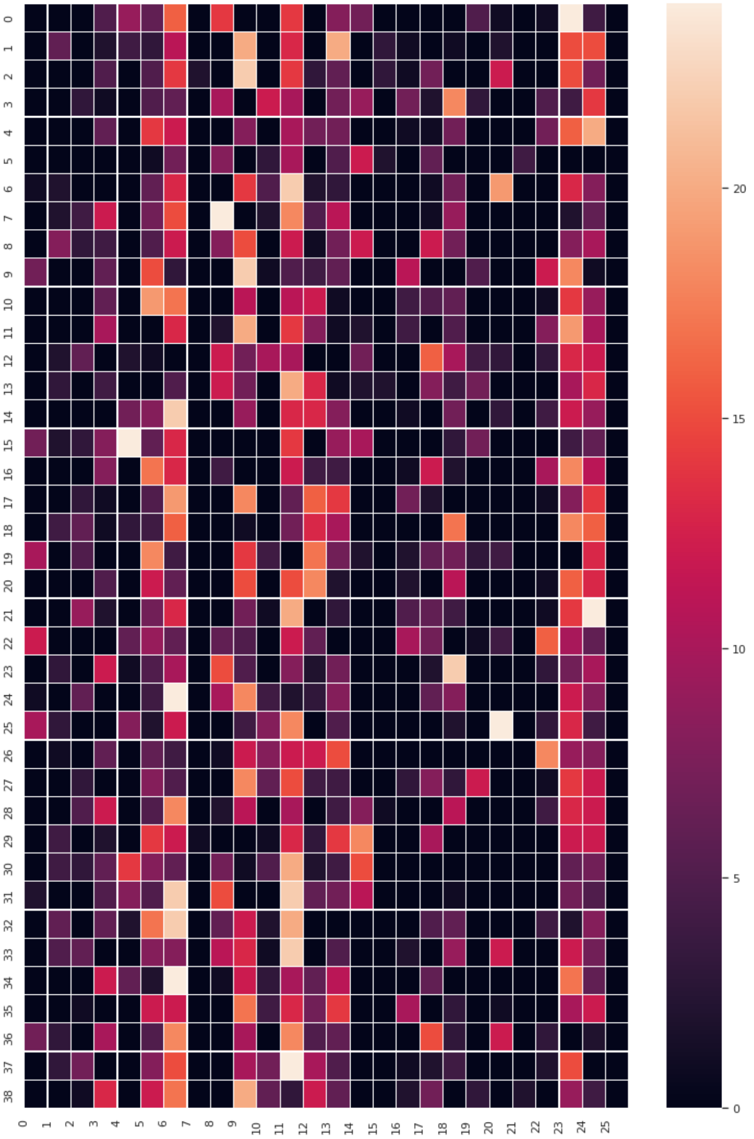


**Figure 3: Visualization of preference matrix (2D)**

- Some critics have averred that a voter need to be forced to "make up his or her mind" by way of expressing a clear reduce desire among any two applicants.
- The language is ordinal, that is why, the method used needs to be ordinal as properly. The majority judgement is ordinal. Methods which are based on sums or averages of points are not ordinal.

## 5 CONCLUSION

The contributions made in this article can be summarized as follows:

- We highlighted the importance of COMSOC, discussed the various voting methods, specially majority judgement method.
- We proposed the use of the majority judgement, a new methods of voting, for ranking the candidates of Eurovision song contest 2021.
- We discussed the results obtained by using the majority judgement method for the selection of winner in the song contest.

The voting techniques discussed and used in this article may find applications in other research problems where ranking or selection is required. Such kind of research aims in bringing together the computational social choice and the ranking/selection problems such as winner selection in art competition etc. In future, we tend to investigate the biased behaviour in voting using some novel techniques of artificial intelligence and deep learning.

## REFERENCES

[1] M. M. Abudy, Y. Mugerman, and E. Shust. The winner takes it all: Investor sentiment and the eurovision song contest. *Journal of Banking & Finance*, 137:106432, 2022.

[2] A. Amarilli, M. L. Ba, D. Deutch, and P. Senellart. Computing possible and certain answers over order-incomplete data. *Theor. Comput. Sci.*, 797:42–76, 2019.

[3] H. Aziz. Computational social choice: Some current and new directions.

[4] M. Balinski. The Majority Judgement : A New Mechanism for Electing and Ranking. pages 257–269.

[5] M. Balinski and R. Laraki. Le jugement majoritaire. *Commentaire*, (2):413–419, 2007.

[6] M. Balinski and R. Laraki. Election by majority judgment: experimental evidence. In *In Situ and Laboratory Experiments on Electoral Law Reform*, pages 13–54. Springer, 2011.

[7] D. Baumeister and T. Hogrebe. On the Complexity of Predicting Election Outcomes and Estimating Their Robustness. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 12802 LNAI:228–244, 2021.

[8] P. J. Boland. Majority systems and the condorcet jury theorem. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 38(3):181–189, 1989.

[9] F. Brandt, J. Hofbauer, and M. Strobel. Exploring the no-show paradox for condorcet extensions using ehrhart theory and computer simulations. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 520–528, 2019.

[10] V. Chakraborty, T. Delemazure, B. Kimelfeld, P. G. Kolaitis, K. Relia, and J. Stoyanovich. Algorithmic Techniques for Necessary and Possible Winners. *ACM/IMS Trans. Data Sci.*, 2(3):1–23, 2021.

[11] V. Chakraborty and P. G. Kolaitis. The Complexity of Possible Winners on Partial Chains. 2020.

[12] L. Chen, P. Xu, and D. Liu. Effect of Crowd Voting on Participation in Crowdsourcing Contests. *J. Manag. Inf. Syst.*, 37(2):510–535, 2020.

[13] L. R. Chernyakhovskaya, A. I. Malakhova, N. O. Nikulina, and V. I. Batalova. Information and analytical collective decision-making support using intelligent technologies. *J. Phys. Conf. Ser.*, 1864(1), 2021.

[14] V. Cho and D. Chan. How social influence through information adoption from online review sites affects collective decision making. *Enterp. Inf. Syst.*, 15(10):1562–1586, 2021.

[15] P. Dey, N. Misra, and Y. Narahari. Frugal bribery in voting. *Theor. Comput. Sci.*, 676:15–32, 2017.

[16] A. A. Dixit and P. G. Kolaitis. *CAvSAT: Answering Aggregation Queries over Inconsistent Databases via SAT Solving*, volume 1. Association for Computing Machinery, 2021.

[17] Z. Dodevska. Computational Social Choice and challenges f voting in multi-agent systems. *Tehnika*, 74(5):724–730, 2019.

[18] Z. A. Dodevska, A. Kovacevic, M. Vukicevic, and B. Delibašić. Two Sides of Collective Decision Making - Votes from Crowd and Knowledge from Experts. *Lect. Notes Bus. Inf. Process.*, 384 LNBIP:3–14, 2020.

[19] W. P. N. dos Reis, G. L. Lopes, and G. S. Bastos. An arrovian analysis on the multi-robot task allocation problem: Analyzing a behavior-based architecture. *Rob. Auton. Syst.*, 144:103839, 2021.

[20] S. Et.al, Ismail. Embedding Crowd-Vote as Knowledge Source to Support Decision Making on University Program Selection. *Turkish J. Comput. Math. Educ.*, 12(3):2120–2128, 2021.

[21] E. Falcó and J. L. Garcia-Lapresta. A distance-based extension of the majority judgement voting system. *Acta Universitatis Matthiae Belii, series Mathematics*, 18:17–27, 2011.

[22] B. Fazzinga, S. Flesca, F. Furfaro, and E. Masciari. Rfid-data compression for supporting aggregate queries. *ACM Trans. Database Syst.*, 38(2):11, 2013.

[23] J. M. Fernandes, L. Geese, and C. Schwemmer. The impact of candidate selection rules and electoral vulnerability on legislative behaviour in comparative perspective. *Eur. J. Polit. Res.*, 58(1):270–291, 2019.

[24] S. Flesca, F. Furfaro, and E. Masciari. On the minimization of xpath queries. *J. ACM*, 55(1):2:1–2:46, 2008.

[25] S. Flesca and E. Masciari. Efficient and effective web change detection. *Data Knowl. Eng.*, 46(2):203–224, 2003.

[26] V. Ginsburgh and A. G. Noury. The eurovision song contest. is voting political or cultural? *European Journal of Political Economy*, 24(1):41–52, 2008.

[27] A. Imber, J. Israel, M. Brill, and B. Kimelfeld. Approval-Based Committee Voting under Incomplete Information. pages 1–23, 2021.

[28] J. Kacprzyk, J. M. Merigó, H. Nurmi, and S. Zadrożny. Multi-agent Systems and Voting: How Similar Are Voting Procedures. *Commun. Comput. Inf. Sci.*, 1237 CCIS:172–184, 2020.

[29] B. Kenig and B. Kimelfeld. Approximate inference of outcomes in probabilistic elections. *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019*, pages 2061–2068, 2019.

[30] B. Kimelfeld, P. G. Kolaitis, and J. Stoyanovich. Computational social choice meets databases. *IJCAI Int. Jt. Conf. Artif. Intell.*, 2018-July:317–323, 2018.

[31] B. Kimelfeld, P. G. Kolaitis, and M. Tibi. Query evaluation in election databases. *Proc. ACM SIGACT-SIGMOD-SIGART Symp. Princ. Database Syst.*, pages 32–46, 2019.

[32] A. Kovacevic, M. Vukicevic, S. Radovanovic, and B. Delibasic. CrEx-Wisdom Framework for Fusion of Crowd and Experts in Crowd Voting Environment – Machine Learning Approach. *Commun. Comput. Inf. Sci.*, 1260 CCIS:131–144, 2020.

[33] J. Lang. Collective Decision Making under Incomplete Knowledge : Possible and Necessary Solutions Jérôme Lang To cite this version : HAL Id : hal-02984842 Collective Decision Making under Incomplete Knowledge : Possible and Necessary Solutions. 2021.

[34] N. Mattei. Closing the loop: Bringing humans into empirical computational social choice and preference reasoning. *IJCAI Int. Jt. Conf. Artif. Intell.*, 2021-January:5169–5173, 2020.

[35] H. Mohajan. Majority judgment in an election with borda majority count. 2011.

[36] F. Mohsin, L. Luo, W. Ma, I. Kang, Z. Zhao, A. Liu, R. Vaish, and L. Xia. Making Group Decisions from Natural Language-Based Preferences. pages 1–21.

[37] A. Pavao. ranky. https://github.com/didayolo/ranky, 2020.

[38] J. Peters. Online Elicitation of Necessarily Optimal Matchings. 2021.

[39] L. Spierdijk and M. Vellekoop. Geography, culture, and religion: Explaining the bias in eurovision song contest voting. *Enschede: Department of Applied Mathematics, University of Twente*, 33, 2006.

[40] D. Stockemer, A. Blais, F. Kostelka, and C. Chhim. Voting in the eurovision song contest. *Politics*, 38(4):428–442, 2018.

[41] A. Umair and E. Masciari. Sentimental and spatial analysis of covid-19 vaccines tweets. *Journal of Intelligent Information Systems*, pages 1–21, 2022.

[42] A. Umair, E. Masciari, and M. H. Habib Ullah. Sentimental analysis applications and approaches during covid-19: a survey. In *25th International Database Engineering & Applications Symposium*, pages 304–308, 2021.

[43] X. Wu, Q. Ye, H. Hong, and Y. Li. Stock Selection Model Based on Machine Learning with Wisdom of Experts and Crowds. *IEEE Intell. Syst.*, 35(2):54–64, 2020.

[44] C. Yan, P. Swaroop, M. O. Ball, C. Barnhart, and V. Vaze. Majority judgment over a convex candidate space. *Oper. Res. Lett.*, 47(4):317–325, 2019.

# QLC: A Quantum-Logic-inspired Classifier

Ingo Schmitt
Brandenburgische Technische Universität Cottbus/Senftenberg
Cottbus, Germany
schmitt@b-tu.de

## ABSTRACT

Besides a good prediction a classifier is to give an explanation how the input data is related to the classification result. There is a general agreement that logic expressions provide a better explanation than other methods like SVM, logistic regression, and neural networks. However, a classifier based on Boolean logic needs to map continuous data to Boolean values which can cause a loss of information. In contrast, we design a quantum-logic-inspired classifier where continuous data are directly processed and the laws of the Boolean algebra are maintained. As a result from our approach we obtain a CQQL condition which provides good insights into the relation of input features to the class decision. Furthermore, our experiment shows a good prediction accuracy.

## CCS CONCEPTS

• **Computing methodologies → Machine learning algorithms**; **Vagueness and fuzzy logic**.

## KEYWORDS

Quantum logic, Classifier, Interpretable AI

**ACM Reference Format:**
Ingo Schmitt. 2022. QLC: A Quantum-Logic-inspired Classifier. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3548785.3548790

## 1 INTRODUCTION AND RELATED WORK

The general classification problem is given by the following components. Let $D$ be a set of objects. Every object $o$ is characterized by its values for $n$ given attributes: $o = (o_1, \dots, o_n)$. Let $Cl = \{class_1, \dots, class_k\}$ be a set of $k$ classes and $m$ be a mapping from $D$ to $Cl$ ($m : D \longrightarrow Cl$). The mapping is typically not explicitly known for all objects and is the learning target. Let $O \subset D$ be a subset of $D$ where for every object the class membership is known. That is, we hold $M = \{(o, m(o)) | o \in O\}$. Let $TR \subset M$ be a set of training objects and $TE = M \setminus TR$ be the test set.

The classification problem is to construct a mapping function $cl : D \longrightarrow Cl$, called a *classifier*, from given training objects $TR$. The classifier should provide a good prediction on $TE$. The *accuracy* of a classifier can be quantified as the fraction of correctly classified objects of all test objects:

$$accuracy = \frac{|\{(o, m(o)) \in TE | m(o) = cl(o)\}|}{|TE|}$$

In the following, we reformulate the initial $k$-class classification problem to $k$ one-class classification problems for $i = 1, \dots, k$:

$$\begin{array}{rcl} m_i : D & \longrightarrow & \{0, 1\} \\ m_i(o) & \mapsto & \begin{cases} 1 & \text{if } m(o) = class_i \\ 0 & \text{otherwise.} \end{cases} \end{array}$$

Thus, we need to determine the classifiers $cl_i : D \longrightarrow \{0, 1\}$ with high accuracy. Next, we will focus on $i = 1$ and will write for short $m$ and $cl$ instead of $m_1$ and $cl_1$.

Besides accuracy, a good classifier $cl$ should provide a good human understanding [3] of the relation between an object $o$ and its corresponding class $m(o)$. The focus of the following work is on human understanding of the relation by means of logic.

There are well-known standard classifier methods like SVM, Bayesian, k-NN, neural network, and decision tree, see [1]. While the listed classifier methods provide fairly good accuracy results for many applications only the decision tree is seen as a classifier providing a good understanding in many cases. Decision tree nodes and edges refer to Boolean conditions and a path corresponds to the conjunction of Boolean conditions. Thus, the decision tree is based on Boolean logic decisions [4]. There is the general agreement that logic-based classifier methods provide a better understanding of the classification process rather than methods without being founded in logic [10]. In the following text we focus on logic-based classifier methods.

Non-classifier approaches for achieving a good understanding of the relation between input objects and binary decisions are *Qualitative Comparative Analysis* QCA and its refinement fsQCA [9]. Both are logic-based and try to find *necessary* and *sufficient* conditions for a binary decision. fsQCA stands for fuzzy set QCA. The logic-based methods decision tree, QCA, and fsQCA use atomic, Boolean conditions on input data that are mainly realized by Boolean comparisons with thresholds. Their resulting Boolean values are combined, evaluated and analyzed. As a general problem, this kind of input mapping of typically real numbers to binary Boolean values can cause a loss of input information. Thus, interactions between input reals can only be expressed on Boolean level but not directly on value level. We will call such classifier methods *input thresholds* methods.

As example, we visualize in Fig. 1 input thresholds for a non-linearly separable problem: a conjunction

$$th_{0.5}(x_1) \wedge th_{0.5}(x_2)$$

where

$$th_\tau(x) = \begin{cases} 1 & \text{if } x \geq \tau \\ 0 & \text{otherwise.} \end{cases}$$

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08.
https://doi.org/10.1145/3548785.3548790

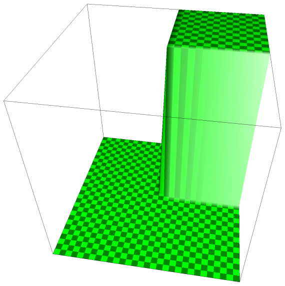


**Figure 1: Boolean Logic: input thresholds in the unit cube** $[0,1]^3$

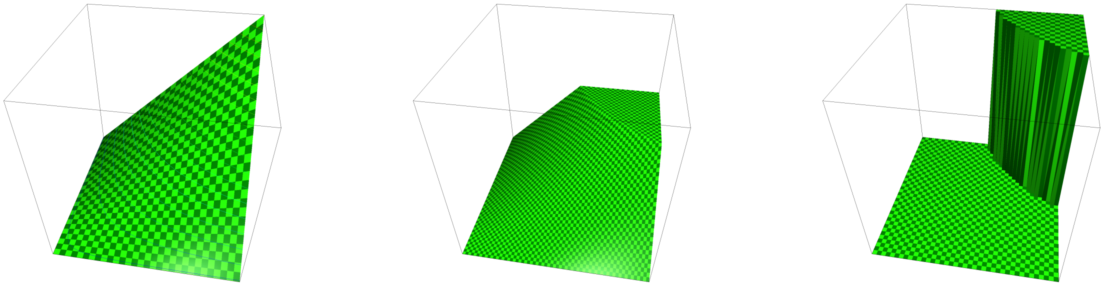


**Figure 2: Output threshold on** $x_1 \cdot x_2$ **(CQQL conjunction): no threshold (left), values below threshold (middle), and applying a threshold on output (right)** $th_{0.5}(x_1 \cdot x_2)$

An input object $(x_1, x_2)$ is represented by a point in a horizontal plane whereas the vertical axis refers to the classification output 0 or 1. Interaction between $x$ and $y$ is expressed by the Boolean conjunction. We see, that input thresholds refer to axis-parallel, vertical cuts of the plane forming blocks which are active (output 1) or inactive (output 0). Modifications of the input thresholds just move block sides along an axis.

In contrast, other classification methods apply a Boolean mapping as a last step for obtaining a class decision. Those methods are typically not based on logic and do not lose information by Boolean input mapping but may suffer from missing human understanding. We will call them *output threshold* methods.

An output threshold on top of a fuzzy conjunction $th_{0.5}(x_1 \wedge x_2) := th_{0.5}(x_1 * x_2)$ is depicted in Fig. 2. It expresses, that high values for $x_1$ and $x_2$ correspond to class 1. In contrast to input thresholds there is no axis-parallel block structure. The output threshold refers to a horizontal cut at a certain height on the vertical axis.

By comparing input thresholds methods with output threshold methods we see the effects of modifying threshold values. Because of the perpendicular cut planes (vertical vs. horizontal), neither output threshold methods subsume input thresholds methods nor vice versa. They express different semantics. That is, no method is, in general, better than the other one with respect to prediction. The method must fit to the the class boundary of the given classification problem.

For output threshold methods we demonstrated a fuzzy logic [2] approach. However, fuzzy logic including t-norm and t-conorm from Zadeh [16] and Łukasievicz [5] suffers from violating important rules of logic. For example, Zadehs max-function violates the law of the excluded third $(x \vee \neg x = 1)$ for $x = 0.5 : \max(0.5, 1 - 0.5) \neq 1$. The product as t-norm, however, violates idempotence $(x \wedge x = x \neq x^2 = x \cdot x)$ for $x \in ]0, 1[$. Thus, fuzzy logic based on those t-norms violates Boolean laws.

In order to overcome the deficiencies of fuzzy logic, we adopt quantum logic and develop a quantum-logic-inspired classifier as an output threshold method. First concepts were published in [6]. The elements of a quantum logic [8] are projectors, which identify subspaces of a Hilbert space. Each projector represents a condition. Based on the subset relation and an operation for negation, the projectors form an orthomodular lattice. Two projectors $p_1$ and $p_2$ are called *commuting* if and only if $p_1 * p_2 = p_2 * p_1$ holds. Projectors being pair-wisely commuting provide a Boolean sublattice [8]. That result from quantum logic is the theoretical foundation of our proposed classifier based on QL conditions. The evaluation of QL conditions deals with input values from the unit interval $[0, 1]$. That is, we assume for the input the existence of $n$ atomic, unary conditions[1] on values $o_i$ of $o = (o_1, \ldots, o_n)$ returning a truth value from the unit interval $[0, 1]$.

For our quantum-logic-based classifier we identify the following characteristics:

- Logic interpretation provides a good understanding and gives us a powerful theory for processing logic expressions.

[1] They correspond to mutually commuting atoms (projectors) of a distributive and hence Boolean QL lattice.

- Avoiding input thresholds supports interactions on the level of continuous values.
- In contrast to fuzzy logic the evaluation of our classification conditions obeys Boolean laws.

Thus, we get both interactions on continuous data and Boolean laws from quantum logic. For example, by applying Boolean laws we obtain in Section 7 the CQQL condition 'G ∨ (A ∧ D ∧ I)' from the PIMA dataset[2] about risk of diabetes: There is a high risk of diabetes in case of a high plasma glucose concentration (G) or in case of a high age (A) together with a strong diabetes pedigree (D) and a high level of Insulin (I).

In following sections we will develop our quantum-logic-inspired classifier. It is based on the concepts of CQQL. After introducing the main concepts of CQQL, we apply them in order to derive from training data weights on minterms (conjunctions of negated or non-negated atomic conditions) of a logic expression in disjunctive normal form. The weights represent the logical expression and relate the input data to the classification result. For a binary classification result we compute an appropriate threshold value on the evaluation result of the logic expression against an input. Last, an experiment demonstrates how our approach can be used to predict a classification result and how to understand the underlying mapping based on logic.

## 2 COMMUTING QUANTUM QUERY LANGUAGE (CQQL)

The quantum-logic-inspired language CQQL (commuting quantum query language) was introduced in [12, 14]. A CQQL condition corresponds to a projector $p$, that is, $p$ is a self-adjoint, idempotent, and linear operator of a Hilbert space [15] and an element of an orthomodular lattice (quantum logic) with meet for the conjunction, join for the disjuction and an orthocomplement for the negation. In that formalism, an object $o$ corresponds to a ket vector $|o\rangle$ of the length of one being constructed by use of the tensor product $|o\rangle := |o_1\rangle \otimes \ldots \otimes |o_n\rangle$. Evaluating a projector $p$ with respect to an object $|o\rangle$ corresponds to a quantum measurement expressed by $\langle o|p|o\rangle \in [0,1]$. Mutually commuting projectors lead to a distributive sublattice and, hence, to a Boolean sublattice [8].

Syntactically, a CQQL condition is a Boolean expression (conjunction, disjunction, negation). We assume $n$ atomic, unary conditions on the $n$ values of an object $o$. Such a condition expresses gradually whether an input value is a high value. It can be easily shown by tensor product construction, see [11], that the $n$ atomic conditions are mutually commuting. Thus, all conditions constructed on those atomic conditions form a Boolean (orthomodular, distributive) lattice.

Above we defined the evaluation of a CQQL condition $e$ against an object $o$ by calculating $\langle o|p_e|o\rangle$ where $|o\rangle$ and $p_e$ are the corresponding ket vector and the corresponding projector, respectively. Actually, from [11, 12, 14] we know that for evaluating conjunction, disjunction, and negation of a CQQL condition we do not need to construct $|o\rangle$ and $p_e$. Instead, we obtain the same evaluation result by applying simple arithmetic operations if the CQQL condition is in a specific normal form (CQQL normal form). Let $atoms(e)$ be the set of atomic conditions involved by a possibly nested condition $e$. The CQQL normal form requires that for each conjunction $e_1 \wedge e_2$ and for each disjunction $e_1 \vee e_2$ (but not for the special case of an exclusive disjunction) the atom sets are disjoint: $atoms(e_1) \cap atoms(e_2) = \emptyset$. If for $e_1 \vee e_2$ the conjunction $e_1 \wedge e_2$ is unsatisfiable in propositional logic then the disjunction is exclusive. We mark each exclusive disjunction by $\dot{\vee}$. The test on unsatisfiablity can be syntactically performed by applying Boolean laws. If a CQQL condition is not in CQQL normal form then it can be syntactically transformed into that normal form by applying Boolean laws [12].

A CQQL condition $e \in CQQL$ in the required normal form is evaluated against an object $o$ by recursively defining

$$eval : CQQL \times D \longrightarrow [0,1] :$$

- Atomic condition: If $e$ is an atomic condition then
$$eval(e, o) \in [0,1]$$
returns the result from applying the corresponding function on $o$.
- Negation: $eval(\neg e, o) = 1 - eval(e, o)$.
- Conjunction: $eval(e_1 \wedge e_2, o) = eval(e_1, o) * eval(e_2, o)$.
- Disjunction (non-exclusive):

$$eval(e_1 \vee e_2, o) = eval(e_1, o) + eval(e_2, o) - eval(e_1, o) * eval(e_2, o).$$

- Exclusive disjunction.:
$$eval(e_1 \dot{\vee} e_2, o) = eval(e_1, o) + eval(e_2, o).$$

For brevity, if the object $o$ is uniquely given from the context then we will simply write $eval(e)$ instead of $eval(e, o)$.

We now extend the expressive power of a CQQL condition by introducing *weighted conjunction* ($e_1 \wedge_{\theta_1,\theta_2} e_2$) and *weighted disjunction* ($e_1 \vee_{\theta_1,\theta_2} e_2$). [13] develops the concept of weights in CQQL from quantum mechanics and quantum logic. Weight variables $\theta_1, \theta_2$ stand for values out of $[0,1]$. A weight $eval(\theta_i) = 0$ means that the corresponding argument has no impact and a weight $eval(\theta_i) = 1$ equals the unweighted case (full impact). We regard every weight variable $\theta_i$ as a 0-ary atomic condition. Before we evaluate a condition with weights we map all weighted conjunctions and disjunctions into an unweighted condition:

$$\begin{array}{rcl}(e_1 \wedge_{\theta_1,\theta_2} e_2) & \rightarrow & ((e_1 \vee \neg\theta_1) \wedge (e_2 \vee \neg\theta_2)) \\ (e_1 \vee_{\theta_1,\theta_2} e_2) & \rightarrow & ((e_1 \wedge \theta_1) \vee (e_2 \wedge \theta_2))\end{array}$$

See the modified example of a diabetes condition from the introduction:

$$e = \mathtt{G} \vee_{1,1} (\mathtt{A} \wedge \mathtt{D} \wedge \mathtt{I}).$$

Both arguments of the conjunction have the maximum weight of 1. Mapping it to the unweighted conjunction yields

$$e = (\mathtt{G} \wedge 1) \vee ((\mathtt{A} \wedge \mathtt{D} \wedge \mathtt{I}) \wedge 1).$$

The value 1 is the neutral element of a conjunction. Thus, we simplify to

$$\mathtt{G} \vee (\mathtt{A} \wedge \mathtt{D} \wedge \mathtt{I}).$$

The condition is in the CQQL normal form and by applying the evaluation rules above we obtain following arithmetic formula for evaluation:

$$eval(e, o) = \mathtt{G} + \mathtt{ADI} - \mathtt{GADI}.$$

[2] https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

Let an object $o$ have the values $(0.3, 0.2, 0.9, 0.4)$ for the attributes $(\mathtt{G}, \mathtt{A}, \mathtt{D}, \mathtt{I})$ then we get:

$$eval(e, o) = 0.3 + (0.2 \cdot 0.9 \cdot 0.4) - (0.3 \cdot 0.2 \cdot 0.9 \cdot 0.4) = 0.3504.$$

So far, we have introduced binary conjunction and disjunction. Since the evaluation of a CQQL condition obeys Boolean laws (commutativity, associativity), we can write $n$-ary disjunction and $n$-ary conjunction as short form for a nested binary operation.

## 3 COMPLETE DISJUNCTIVE NORMAL FORM

For a certain classification problem, we want to find a matching CQQL condition $e$ together with a well chosen output threshold value $\tau$

$$cl_e^\tau(o) = th_\tau(eval(e, o)) \text{ where } th_\tau(x) = \begin{cases} 1 & \text{if } x \geq \tau \\ 0 & \text{otherwise.} \end{cases}$$

From the laws of the Boolean algebra we know that every condition $e$ can be expressed in the complete disjunctive normal form, that is, every condition is equivalent to a subset of $2^n$ minterms. We assume for each of the $n$ object attributes exactly one atomic condition $c_j$. The minterm subset relation for a condition can be expressed by use of minterm weights $\theta_i \in \{0, 1\}$:

$$\begin{aligned} e := \dot{\bigvee}_{i=1}^{2^n} minterm_{i,\theta_i} &= \dot{\bigvee}_{i=1}^{2^n} minterm_i \wedge \theta_i && (1) \\ minterm_i &= \bigwedge_{j=1}^{n} c_{ij} && (2) \\ c_{ij} &= \begin{cases} c_j & \text{if } (i-1)\ \&\ 2^{j-1} > 0 \\ \neg c_j & \text{otherwise.} \end{cases} && (3) \end{aligned}$$

The symbol '&' stands for bitwise and, $i - 1$ is considered as a binary number and $j$ as a bit position.

Notice that the disjunction of two different complete minterms is always exclusive. Thus, $e$ is in CQQL normal form and its evaluation against object $o$ yields

$$\begin{aligned} eval(e, o) &= \sum_{i=1}^{2^n} \theta_i \prod_{j=1}^{n} c_{ij}^o \text{ where} && (4) \\ c_{ij}^o &= \begin{cases} eval(c_j, o) & \text{if } (i-1)\ \&\ 2^{j-1} > 0 \\ 1 - eval(c_j, o) & \text{otherwise.} \end{cases} && (5) \end{aligned}$$

## 4 EXTRACTION OF MINTERMS

Next, we will extract a CQQL condition $e$ in complete disjunctive normal form from training data. We have to find the weight value $\theta_i$ for every minterm $i$. The starting point is a set $TR$ of $(x, y)$ pairs where $x$ refers to a training object from $O$ with $x = (x_1, \ldots, x_n)$ and $y = m(x) \in \{0, 1\}$.

As preparatory step we map all $x_i$ from $M$ to $x_i' \in [0, 1]$ by using a monotonic mapping function for realizing $eval(c_j, x)$. We interpret $x_i'$ as a gradual truth value telling us how high the attribute value $x_i$ is. It si similar to the concept of a linguistic label in fuzzy logic. The mapping function can be linear $x_i' = (x_i - min_i)/(\max_i - \min_i)$. As an alternative mapping function we may follow a probabilistic approach and take $x_i' = P(X_i \leq x_i) = \int_{-\infty}^{x_i} f(x)dx$ where $f(x)$ is a density function. In the following, we assume that every $x$ from $TR \cup TE$ is an element of the hyper-cube $[0, 1]^n$.

A good classifier is one with high accuracy. Therefore, we maximize the accuracy of condition (1) depending on the minterm weights $\theta_i$ based on $TR$.

Before we start, we have to adapt the formula for accuracy to our CQQL scenario. That is, we regard $eval(e, x) \in [0, 1]$ as evaluation result of a condition $e$ where a value near to 1 corresponds to *true* and a value near to 0 corresponds to *false*. For a given $(x, y)$-pair and the evaluation result of a CQQL condition $e$ we can distinguish four cases of a confusion matrix:

| | $y$ | $\neg y$ |
|---|---|---|
| $eval(e, x)$ | $y \wedge e$ | $\neg y \wedge e$ |
| $eval(\neg e, x)$ | $y \wedge \neg e$ | $\neg y \wedge \neg e$ |

The cases on the diagonal ($y \wedge e$ is the correct prediction in case of acceptance, also called correct alarm, and $\neg y \wedge \neg e$ is the correct prediction in case of rejection, also called correct rejection) refer to the correct results. Accuracy $acc$ for a continuous evaluation result $eval(e, x) = 1 - eval(\neg e, x)$ can now be computed over the two correct cases $y * eval(e, x) + (1 - y)(1 - eval(e, x))$. Summing up over all training data yields:

$$\begin{aligned} acc &= \sum_{(x,y) \in TR} (y * eval(e, x) + (1 - y)(1 - eval(e, x))) \\ &= \sum_{(x,y) \in TR} (eval(e, x) \cdot (2y - 1) + 1 - y) \\ &= \sum_{(x,y) \in TR} eval(e, x) \cdot (2y - 1) + \sum_{(x,y) \in TR} (1 - y) \\ &= \sum_{(x,y) \in TR} \left( \sum_{i=1}^{2^n} \theta_i \cdot \prod_{j=1}^{n} c_{ij}^x \right) \cdot (2y - 1) + \sum_{(x,y) \in TR} (1 - y) \\ &= \sum_{i=1}^{2^n} \theta_i \sum_{(x,y) \in TR} \left( (2y - 1) \cdot \prod_{j=1}^{n} c_{ij}^x \right) + \sum_{(x,y) \in TR} (1 - y). \end{aligned}$$

We see that accuracy shows a linear dependence on the minterm weights $\theta_i$ for fixed $TR$-pairs. The first derivative provides the constant gradient on $\theta_i$:

$$\frac{\partial acc}{\partial \theta_i} = \sum_{(x,y) \in TR} (2y - 1) \cdot \prod_{j=1}^{n} c_{ij}^x.$$

Because of $y \in \{0, 1\}$ we reformulate the gradient to:

$$\frac{\partial acc}{\partial \theta_i} = \sum_{(x,1) \in TR} \prod_{j=1}^{n} c_{ij}^x - \sum_{(x,0) \in TR} \prod_{j=1}^{n} c_{ij}^x.$$

For maximizing accuracy a minterm weight $\theta_i$ should have the value 1 if $\frac{\partial acc}{\partial \theta_i} > 0$ and 0 otherwise:

$$\theta_i = \begin{cases} 1 & \text{if } \sum_{(x,1) \in TR} \prod_{j=1}^{n} c_{ij}^x > \sum_{(x,0) \in TR} \prod_{j=1}^{n} c_{ij}^x \\ 0 & otherwise. \end{cases} \qquad (6)$$

In other words, for the decision whether a minterm should be active (having value 1) or not (having value 0) it is sufficient to compare the impact of positive training data $(x, 1) \in TR$ against the impact of the negative training data $(x, 0) \in TR$ on minterm $i$. Be aware that the decision depends on the relative number of positive training objects. Therefore, let $\gamma_1 = \sum_{(x,1)\in TR} 1$ and $\gamma_0 = \sum_{(x,0)\in TR} 1$ be the number of positive and negative training objects, respectively. The fraction of negative objects is then given by $\gamma = \frac{\gamma_0}{\gamma_1+\gamma_0}$. In the uneven case ($\gamma \neq 1/2$) we compensate the effect on the minterm weight decision by:

$$\theta_i = \begin{cases} 1 & \text{if } \gamma \cdot E_i > (1-\gamma) \cdot N_i \\ 0 & otherwise \end{cases} \tag{7}$$

where

$$E_i = \sum_{(x,1)\in TR} \prod_{j=1}^{n} c_{ij}^x \qquad N_i = \sum_{(x,0)\in TR} \prod_{j=1}^{n} c_{ij}^x.$$

We obtained that decision rule by maximizing accuracy where correct alarm and correct rejection are given the same weight. However, in some real life scenarios correct alarms should have a different weight than correct rejections. Such kind of *weighted accuracy* can be expressed by:

$$\lambda * y * eval(e,x)) + (1-\lambda) * (1-y) * (1 - eval(e,x))$$

where $\lambda$ is the weight for correct alarms. The trade-off between good recall and good precision leads to the decision rule

$$\theta_i = \begin{cases} 1 & \text{if } \gamma \cdot \lambda \cdot E_i > (1-\gamma) \cdot (1-\lambda) \cdot N_i \\ 0 & otherwise \end{cases}. \tag{8}$$

## 5 STABLE MINTERM WEIGHTS

Following the minterm decision rule (8) we decide whether a minterm is active or inactive. For some minterms the decision is very clear. However, the decision is not clear if the left term is very close to the right term ($\gamma \cdot \lambda \cdot E_i \approx (1-\gamma) \cdot (1-\lambda) \cdot N_i$). We call such kind of minterms *unstable* because of adding a single new training object may change the decision. Instable minterms have a low impact on the result. We are interested in stable minterms. For measuring stability we compute the ratio $\rho_i$ of the left side to the sum of both sides of a minterm *i*:

$$\rho_i = \frac{\gamma\lambda E_i}{\gamma\lambda E_i + (1-\gamma)(1-\lambda)N_i} \in [0,1]. \tag{9}$$

A value for $\rho_i$ close to 1/2 means an unstable minterm *i*, a value near to 1 means a stable active minterm *i*, and a value near to 0 means a stable inactive minterm *i*.

The question is now, should instable minterms be active or inactive? We propose to sort all minterms by their values for $\rho_i$ and choose a $\rho$-threshold $\theta_\rho$ out of them that provides a sufficient accuracy and a good compactness of condition *e*. The modified minterm decision rule is:

$$\theta_i = \begin{cases} 1 & \text{if } \frac{\gamma\lambda E_i}{\gamma\lambda E_i + (1-\gamma)(1-\lambda)N_i} > \theta_\rho \\ 0 & otherwise \end{cases}. \tag{10}$$

## 6 FINDING OUTPUT THRESHOLD

After applying our decision rule (10) we obtain the formula:

$$e = \bigvee_{i:\theta_i=1} \bigwedge_{j=1}^{n} c_{ij}.$$

```
minterm_eval(i,o,n)
  value := 1
  for j in range(n)
    if (i-1)&$2^j$
      value := value * eval($c_j$,o)
    else
      value := value * (1 - eval($c_j$,o))
  return value
```

**Figure 3: minterm_eval**

```
get_weight(i,TR,n,$\lambda$,$\theta_\rho$)
  E := 0  N := 0  $\gamma_1$ := 0 $\gamma_0$ :=0
  for (x,y) in TR
    if y == 1
      $\gamma_1$ := $\gamma_1$ + 1
      E := E + minterm_eval(i,x,n)
    else
      $\gamma_0$ := $\gamma_0$ + 1
      N := N + minterm_eval(i,x,n)
  $\gamma = \frac{\gamma_0}{\gamma_0+\gamma_1}$
  $\rho = \frac{\gamma\lambda E}{\gamma\lambda E+(1-\gamma)(1-\lambda)N}$
  if $\rho > \theta_\rho$
    return 1
  return 0
```

**Figure 4: get_weight**

```
object_eval(o,n,$\{\theta_i\}$)
  value := 0
  for i in range(1,$2^n$+1)
    if $\theta_i$ == 1
      value := value + minterm_eval(i,o,n)
  return value
```

**Figure 5: object_eval**

```
class(o,n,,$\{\theta_i\}$,$\tau$)
  if object_eval(o,n,$\{\theta_i\}$)$\geq \tau$
    return 1
  return 0
```

**Figure 6: class**

Its evaluation against an object *o* returns a continuous value from the unit interval: $eval(e,o) \in [0,1]$. As a last step, we have to find the output threshold value $\tau$ for

$$cl_e^\tau(o) = th_\tau(eval(e,o)). \tag{11}$$

Let $\min_1 = \min_{(x,1)\in TR} eval(e,x)$ be the smallest evaluation result of the positive training objects and $\max_0 = \max_{(x,0)\in TR} eval(e,x)$ the highest result of the negative training objects. In case of $\max_0 <$

**Table 1:** *accuracy*$(e, \tau, \_)$ **for different** $\theta_\rho$**-values (**$\theta_\rho$ **and accuracy in percent)**

| $\theta_\rho$ | $\tau$ | accuracy on TR | accuracy on TE | #minterms |
|---|---|---|---|---|
| 50 | 0.61 | 80.6 | 77.41 | 112 |
| 55 | 0.522 | 80.3 | 74.19 | 89 |
| 60 | 0.474 | 80.3 | 80.64 | 72 |
| 65 | 0.396 | 80.3 | 79.03 | 57 |
| 70 | 0.242 | 80.3 | 77.41 | 39 |
| 75 | 0.167 | 80.6 | 74.19 | 24 |
| 80 | 0.066 | 79.7 | 69.35 | 9 |
| 85 | 0.0103 | 77.6 | 67.74 | 2 |

$\min_1$ positive objects and negative objects are well separated and we set $\tau = (\max_0 + \min_1)/2$.

Otherwise, we have to choose a $\tau$ value from the interval

$$\tau \in [\min_1, \max_0].$$

In order to find a threshold which maximizes discrete accuracy we use the training objects from $TR$:

$$\tau = \operatorname*{arg\,max}_{\substack{(x,\_) \in TR, \\ \tau_x := eval(e,x), \\ \tau_x \in [\min_1, \max_0]}} accuracy(e, \tau_x, TR) \quad (12)$$

where

$$accuracy(e, \tau_x, TR) = \frac{|\{(x, y) \in TR | y = cl_e^{\tau_x}(x)\}|}{|TR|}.$$

For evaluating the power of prediction the classifier needs to be checked against the test set $TE$:

$$accuracy(e, \tau, TE) = \frac{|\{(x, y) \in TE | y = cl_e^{\tau}(x)\}|}{|TE|}.$$

The extracted CQQL condition $e$ can now be presented to the user and gives an understanding of the logical connection between input and class decision. A condition in disjunctive normal form is often hard to understand. Due to the fact, that for our approach the rules of the Boolean algebra hold the condition can be brought into another syntactical form. If the number of attributes is small then the Quine–McCluskey algorithm [7] may be applied to simplify condition $e$. Notice, because of having $2^n$ minterms our approach has exponential time and space complexity on the number of attributes.

Besides giving an understanding the condition should make good predictions on the test set. It can happen, that the described approach suffers from overfitting. In that case, a more compact condition may improve understanding and accuracy on test data. A starting point for finding a more compact condition is to analyze single minterms of the simplified disjunctive normal form and single maxterms of the simplified conjunctive normal form. Next section will demonstrate steps for finding a compact CQQL condition.

The derived formulas can be easily implemented. The algorithm in Figure 3 implements Equation 2, the algorithm in Figure 4 implements Equation 10, the algorithm in Figure 5 implements Equation 4, and the algorithm in Figure 6 implements Equation 11.

## 7 EXPERIMENT

We demonstrate the power of our quantum-logic-inspired classifier by use of the classification problem PIMA. PIMA is a set of data about diabetes of some selected people from India. The data set $M$ contains values for the eight attributes: pregnancies (P), glucose (G), blood pressure (BP), skin thickness (S), insulin (I), BMI, diabetes pedigree function (D), and age (A) as well as the information whether diabetes occured or not. The challenge is to understand the relation between the eight attributes and the occurrence of diabetes and to enable a good diabetes prediction. PIMA contains data about 768 people. However, the data is complete only for 392 people ($|M| = 392$). We partition $M$ into training data $TR$ with $|TR| = 330$ and test data $TE$ with $|TE| = 62$. For further processing the input values of the eight attributes are mapped to the unit cube $[0, 1]^8$ using the density-function-based mapping.

Applying Equation 9 to $TR$ yields the $\rho$-values for $2^8 = 256$ minterms. For selecting the best set of minterms we try out different $\theta_\rho$ thresholds, compute the respective best output threshold $\tau$, measure discrete accuracies on $TR$ and $TE$ and count the number of active minterms, see Tab. 1. For the consecutive discussion, we choose $\theta_\tau = 0.65$ and obtain $e$ in simplified disjunctive normal form, see Tab. 2 left. In Tab. 2 right we see $e$ after its transformation to the simplified concjunctive normal form.

Formula $e$ with discrete accuracy of 79% is very complex and the starting point for understanding the classifier. We search for simpler formulas $f$ and are especially interested in two kinds of formulas when we regard $e$ as condition in propositional logic:

- $f$ is *necessary* ($e \implies f$): All input data that are classified by $e$ as class members satisfy $f$. That is, with respect to $e$, false rejections of $f$ cannot occur and thus recall is 100%.
- $f$ is *sufficient* ($f \implies e$): All input data that satisfy $f$ also hold $e$. That is, with respect to $e$, false positives of $f$ cannot occur and thus precision is 100%.

Please notice, that all minterms of $e$ are sufficient formulas and all maxterms are necessary formulas.

In Table 3 we compare different formulas $f$ with formula $e$. We count the number of minterms of $e \wedge f$ (11), $e \wedge \neg f$ (10), $\neg e \wedge f$ (01), and $\neg e \wedge \neg f$ (00) with $00 + 01 + 10 + 11 = 2^8$ and compute the respective values for precision, recall, and accuracy on the number of shared minterms. At first, we try out every single attribute. We see, that glucose (G) plays with accuracy of 69% an important role. Next, we try out attributes in combination. The first three maxterms in Tab. 2 right are very compact and necessary formulas. Therefore, recall ist 100% but precision is low. In the next section of Table 3, we test simple conjunctions of the three attributes G, BMI, and A. They are neither necessary nor sufficient but show

**Table 2: Minterms (left) and maxterms (right) of $e$ in dnf and cnf, respectively; 0 stands for negated atom, 1 for non-negated atom and an empty cell for no-care**

minterms

| P | G | BP | S | I | BMI | D | A |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | | 1 | | 1 |
| 1 | 1 | | | | 1 | 1 | 1 |
| 1 | | | | 1 | 1 | 1 | 1 |
| | 1 | 1 | | 1 | | 1 | 1 |
| | 1 | 1 | 1 | | | 1 | 1 |
| | 1 | 1 | 1 | | 1 | 1 | |
| | 1 | | 1 | 1 | 1 | 1 | |
| 1 | 1 | 1 | 1 | 1 | | | |
| | 1 | 1 | 0 | | 1 | | 1 |
| | 1 | 0 | | 1 | 1 | | 1 |
| 1 | 1 | 0 | | 1 | | | 1 |
| | 1 | | 1 | 1 | 0 | | 1 |
| 0 | 1 | | 1 | | 1 | 1 | |
| 1 | | 1 | 1 | 1 | | 1 | 1 |
| 1 | 1 | 1 | | 1 | 1 | 1 | |
| | | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | | 1 | 1 | 1 | |

maxterms

| P | G | BP | S | I | BMI | D | A |
|---|---|---|---|---|---|---|---|
| | 1 | | | | | | 1 |
| | 1 | | | | | 1 | |
| | 1 | | | 1 | | | |
| | | 1 | | | 1 | | 1 |
| | | | | 1 | 1 | | 1 |
| 1 | | | | | 1 | | 1 |
| | | | 1 | | 1 | | 1 |
| | | 1 | | | | 1 | 1 |
| | | | | 1 | | 1 | 1 |
| | | | 1 | | | 1 | 1 |
| | | | 1 | 1 | | | 1 |
| | 1 | 1 | | | 1 | | |
| | | 1 | | 1 | 1 | | |
| | | | | 1 | 1 | 1 | |
| | 1 | | 1 | | 1 | | |
| | | | 1 | 1 | 1 | | |
| | | 1 | | 1 | | 1 | |
| 1 | 1 | | 1 | | | | |
| 1 | 1 | 0 | | | | | |
| 1 | | 1 | 1 | | 1 | | |
| 1 | | 1 | 1 | 1 | | | |
| 0 | | 1 | | 1 | | | 1 |
| 0 | | 1 | 1 | | | | 1 |
| 1 | | 0 | 1 | | | | 1 |
| | | 0 | 1 | | 1 | 1 | |
| 1 | | 0 | 0 | | 0 | 1 | |

good accuracy on minterms. In the last section, we combine the tree maxterms conjunctively into one formula and obtain again a necessary formula with better accuracy than the single maxterms.

So far we have tested $f$ against $e$ based on the numbers of shared minterms but $e$ itself shows an accuracy of 79% against the test data. Therefore, we have to test the formulas $f$ from Table 3 against our data set $M = TR \cup TE$ and obtain Table 4. The measurements differ from the measursents in Table 3 since for Table 4 new output threshold values $\tau$ for each formula $f$ are chosen by maximizing accuracy[3]. Interestingly, we see the last formula $\mathtt{G} \vee (\mathtt{A} \wedge \mathtt{D} \wedge \mathtt{I})$ being very compact. Its accuracy is not less than that for $e$. It looks as if we have not derived the optimal formula $e$. However, please be aware that equation 9 is based on a continuous accuracy whereas in Table 4 discrete accuracy was calculated.

In literature[4] the data set PIMA was used to build a logistic regression classifier with accuracy 73%, a SVM classifier with an accuracy of 75%, and a random forest classifier with an accuracy of 88%. Only the last one shows a better accuracy than $\mathtt{G} \vee (\mathtt{A} \wedge \mathtt{D} \wedge \mathtt{I})$. However, our formula $\mathtt{G} \vee (\mathtt{A} \wedge \mathtt{D} \wedge \mathtt{I})$ is very compact and can be interpreted very easily: There is a high risk of diabetes in case of a high plasma glucose concentration (G) or in case of a high age (A) together with a strong diabetes pedigree (D) and a high level of Insulin (I).

## 8 CONCLUSIONS

In our paper we propose a classifier based on quantum logic. In contrast to Boolean logic (input thresholds) quantum logic can directly deal with continuous data, which is typically available in many classification scenarios. Other than fuzzy logic quantum logic is based on a sound theoretical theory and, if some restrictions are respected, it obeys the rules of Boolean algebra. Our quantum-logic-inspired classifier is a tool to interpret the mapping between classification input and output by means of logic. The diabetes experiment demonstrates a good quality of prediction and of explaining the relation between input and class decision. We used rules of Boolean algebra in order to deal with CQQL conditions as same as with propositional logic in order to find a compact condition with good prediction and a good understanding.

However, our approach is based on the disjunctive normal form where the number of minterms explodes with the number of attributes. Thus, we are restricted to classification problems with a small number of attributes.

[3] We are free to modify $\tau$ values in order to improve precision at the expense of recall or vice versa.

[4] https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

**Table 3: Minterm comparisons of formulas $f$ against formula $e$ derived from $\theta_\rho = 0.65$**

| $f$ | precision | recall | accuracy | 00 | 01 | 10 | 11 |
|---|---|---|---|---|---|---|---|
| G | 40 | 89 | 67 | 122 | 6 | 77 | 51 |
| BMI | 31 | 70 | 59 | 111 | 17 | 88 | 40 |
| A | 35 | 79 | 63 | 116 | 12 | 83 | 45 |
| D | 31 | 70 | 59 | 111 | 17 | 88 | 40 |
| P | 27 | 60 | 54 | 105 | 23 | 94 | 34 |
| BP | 26 | 58 | 54 | 104 | 24 | 95 | 33 |
| S | 28 | 63 | 56 | 107 | 21 | 92 | 36 |
| I | 33 | 74 | 61 | 113 | 15 | 86 | 42 |
| A ∨ G | 30 | 100 | 47 | 64 | 0 | 135 | 57 |
| D ∨ G | 30 | 100 | 47 | 64 | 0 | 135 | 57 |
| I ∨ G | 30 | 100 | 47 | 64 | 0 | 135 | 57 |
| G ∧ A ∧ BMI | 78 | 44 | 85 | 192 | 32 | 7 | 25 |
| G ∧ A | 61 | 68 | 83 | 174 | 18 | 25 | 39 |
| G ∧ BMI | 55 | 61 | 80 | 177 | 22 | 29 | 35 |
| A ∧ BMI | 47 | 53 | 76 | 165 | 27 | 34 | 30 |
| G ∨ (A ∧ D ∧ I) | 40 | 100 | 66 | 112 | 0 | 87 | 57 |

**Table 4: Evaluation formulas $f$ against $TR \cup TE$ with new threshold**

| $f$ | precision | recall | accuracy | 00 | 01 | 10 | 11 |
|---|---|---|---|---|---|---|---|
| G | 81 | 42 | 77 | 249 | 76 | 13 | 54 |
| BMI | 53 | 19 | 68 | 240 | 105 | 22 | 25 |
| A | 64 | 36 | 72 | 235 | 83 | 27 | 47 |
| D | 61 | 15 | 69 | 249 | 110 | 13 | 20 |
| P | 70 | 32 | 73 | 244 | 88 | 18 | 42 |
| BP | 66 | 15 | 69 | 252 | 111 | 10 | 19 |
| S | 53 | 23 | 68 | 235 | 100 | 27 | 30 |
| I | 53 | 66 | 69 | 185 | 44 | 77 | 86 |
| A ∨ G | 73 | 59 | 79 | 234 | 53 | 28 | 77 |
| D ∨ G | 77 | 42 | 77 | 246 | 75 | 16 | 55 |
| I ∨ G | 69 | 42 | 74 | 237 | 75 | 25 | 55 |
| G ∧ A ∧ BMI | 75 | 48 | 77 | 241 | 68 | 21 | 62 |
| G ∧ A | 72 | 55 | 78 | 235 | 58 | 27 | 72 |
| G ∧ BMI | 69 | 52 | 77 | 232 | 62 | 30 | 68 |
| A ∧ BMI | 64 | 35 | 72 | 237 | 85 | 25 | 45 |
| G ∨ (A ∧ D ∧ I) | 77 | 58 | 80 | 240 | 55 | 22 | 75 |

In further work, we will design algorithms that do not suffer from the exponential number of minterms. Furthermore, our approach has to be tested against more classification scenarios.

## REFERENCES

[1] Charu C Aggarwal. 2015. *Data mining: the textbook.* Springer.

[2] Radim Belohlavek, Rudolf Kruse, and Christian Moewes. 2011. Fuzzy logic in computer science. In *Computer Science*. Springer, 385–419.

[3] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

[4] Alex A Freitas. 2014. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter* 15, 1 (2014), 1–10.

[5] Robin Giles. 1976. Łukasiewicz logic and fuzzy set theory. *International Journal of Man-Machine Studies* 8, 3 (1976), 313–327.

[6] Eyke Hüllermeier and Ingo Schmitt. 2014. Non-Additive Utility Functions: Choquet Integral versus Weighted DNF Formulas. In *German-Japanese Interchange of Data Analysis Results*. Springer, 115–123.

[7] Edward J McCluskey. 1956. Minimization of Boolean functions. *The Bell System Technical Journal* 35, 6 (1956), 1417–1444.

[8] Peter Mittelstaedt. 1976. Quantum logic. In *PSA 1974*. Springer, 501–514.

[9] Benoit Rihoux and Charles C Ragin. 2008. *Configurational comparative methods: Qualitative comparative analysis (QCA) and related techniques.* Sage Publications.

[10] Stuart Russell and Peter Norvig. 2020. *Artificial Intelligence: A Modern Approach (4th Edition).* Pearson. http://aima.cs.berkeley.edu/

[11] Ingo Schmitt. 2006. *Quantum query processing: unifying database querying and information retrieval.* Citeseer.

[12] Ingo Schmitt. 2008. Qql: A db&ir query language. *The VLDB journal* 17, 1 (2008), 39–56.

[13] Ingo Schmitt. 2019. Incorporating Weights into a Quantum-Logic-Based Query Language. In *Quantum-Like Models for Information Retrieval and Decision-Making.* Springer, 129–143.

[14] Ingo Schmitt and Daniel Baier. 2013. Logic based conjoint analysis using the commuting quantum query language. In *Algorithms from and for Nature and Life.* Springer, 481–489.

[15] Nicholas Young. 1988. *An introduction to Hilbert space.* Cambridge university press.

[16] Lotfi A Zadeh. 1988. Fuzzy logic. *Computer* 21, 4 (1988), 83–93.

# An Online MCQ sub-system for CrsMgr

Maria G. Ratcheva
Concordia University
Montreal, Canada

Reethu Navale
Concordia University
Montreal, Canada

Bipin C. Desai
Concordia University
Montreal, Canada

## Abstract

The current pandemic has led to increased use of online learning and calls for innovative self-learning techniques. Since contact with educators is limited, students are required to become more self-reliant. This endeavour includes using self-assessment tools to measure the learning progress and uncover the areas for further studies. In this paper, we focus on a system that automatically generates various types of questions from the recommended course material. The system applies the most recent machine learning techniques, such as transfer learning, natural language generation methods and finding semantic similarity. We propose a human-in-the-loop approach where the instructor can provide his guidance. Our system would help students to calibrate themselves in a typical remote learning environment.

***CCS Concepts:*** • **Computing methodologies** → **Machine learning**; • **Information systems** → **Data management systems**; • **Human-centered computing** → *Visualization*.

***Keywords:*** Multiple-Choice Question (MCQ), Study quiz, NLP, NLG

**ACM Reference Format:**
Maria G. Ratcheva, Reethu Navale, and Bipin C. Desai. 2022. An Online MCQ sub-system for CrsMgr. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3548785.3548789

## 1 Introduction

CrsMgr [34] is an online system used for over a dozen years to support the administrative and pedagogical processes of university-level teaching. It has also been used to transform traditional "final" exams into multiple online quizzes combining multiple-choice questions (MCQ) and traditional written answers to text questions - the answers could be submitted to the system. This shift in the examination practices is related to the need for prompt examinations, thus, avoiding study delays until the end of the course. Online assessments have become popular among educational institutions, proficiency examiners, and corporations. As a result of the pandemic, most university courses moved online and the evaluation using online evaluation became necessary. One of the issues we discovered was that many students were not used to online quizzes and did not do well. Moreover, the topics for the tests were limited to the most recent lessons. We realized that CrsMgr's online quiz functionality could easily be expanded to reflect the synchronous nature of the online classes.

The motivation of this project is to create on-demand practice quizzes for the students using the required texts and class notes. The generated questions are an amalgamation of multiple-choice questions with one or several correct answers, true or false questions, fill-in-the-blanks questions by choosing appropriate values, and interrogative questions extracted from the learning material. The application uses natural language processing (NLP) and machine learning (ML) techniques. The input source consists of course materials and class notes posted on CrsMgr. Additional inputs are the source of textbooks written by one of the authors. These textbooks are copy-righted and available from the university library and other sources [24, 25]. Furthermore, our system aims to automatically evaluate and provide feedback on the test results, thus helping the students prepare for the real quizzes.

## 2 Present Developments

The manual creation of multiple-choice questions for online quizzes is a cumbersome and time-consuming task for the instructors. On the other hand, the automated generation of questions and answers is challenging and extremely broad, as it involves solving several sub-problems. Many researchers and top organizations are attracted to building online question and answering and assessment applications.

A myriad of online assessment tools, commercial or free, try to provide useful features for online training. Socrative is an interactive student response system that empowers teachers to engage their classrooms through quizzes and pools via smartphones and tablets [13]. ProProfs (Quiz Maker) is another online quiz creation app where users can select from a large library of questions organized by topic or upload the questions from an excel file [10]. Google Forms can also help users manually create quizzes or select a quiz template from many public templates. The advantage is that it works with other Google Apps, so the quiz can be sent to students via

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548789

their mail or embedded in a Google Site. ClassMarker is a secure, professional web-based quiz maker, including instant grading and saving hours of paperwork [1]. Another online application that provides pre-built quiz questions is ThatQuiz [16]. It has built-in quizzes for math, science, language arts, and social studies, which are adjustable in difficulty and length. ExamTime Quizzes [3], Testmoz [15], Gnowledge [4], Online Quiz Creator [9], GoToQuiz [5], QuizStar [12], Survey Anyplace [14], Mentimeter [8], and Edmodo [2] are some more of the many online quiz creation applications.

Although these systems are helpful with the entire examination process, the random selection of questions, the timing and the instant grading, the problem is that the user has to create the questions manually. There are some options to upload questions using Excel templates or select them from a library of pre-built questions, but it is still a time-consuming process.

## 3 CrsMgr Quiz Generation Sub-System

Our quiz generation system is an extension of CrsMgr. Questions are generated from the passages the instructors select from the course materials (lecture notes or textbook excerpts). The instructor has also the option to upload the corresponding pdf file. The generated questions can be classified into multiple choice (MCQ), true/false, fill-in-the-blanks, and questions with an interrogative wh-word (What /Who /When /Why /How). The application uses machine learning models to generate the questions, the corresponding correct answer, as well as the incorrect answers, known as distractors. The obtained question-answers sets are then stored in a database as a question bank for future use. Unlike some of the systems that have pre-built questions, our subsytem for CrsMgr generates the quiz automatically based on the course material. Students could use the same portal for self-learning and all other activities related to the course; they do not need to switch to a different portal.

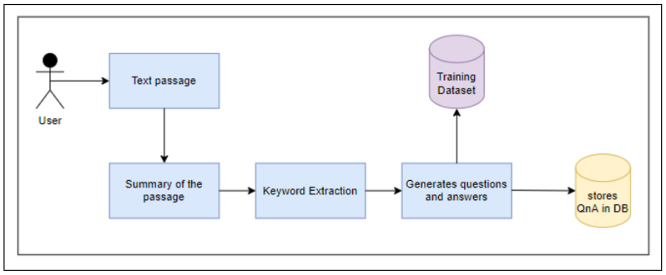


**Figure 1.** The architecture of CrsMgr for generation of practice quiz.

Figure 1. explains the quiz generation system in details. In the first step, the user enters the passage in the text area to generate the questions. Then, the application creates a summary of the text and extracts the keywords from the summary text. The application will then select the sentences that the keywords represent and generate the question. Along with the correct answer, the incorrect answers are also generated. The application uses WordNet [27], a large lexical database of semantically similar words, combined with NLP techniques to produce the distractors.

### 3.1 Multiple Choice Questions (MCQ)

MCQ is a form of objective assessment in which respondents are asked to select the correct answer from the choices offered. The multiple-choice format is the most frequently used in educational testing. Edward Lee Thorndike worked on an early scientific approach to test students, and his assistant Benjamin D. Wood developed the multiple-choice test [17].

In the mid-20th century, MCQ's popularity increased. Multiple choice question types are chosen because they are affordable for testing many students. The drawback is that students can take a guess when answering the question rather than genuinely understand the concept. Despite all the flaws, MCQs are popular because they are easy to create, score and analyze [32].

MCQs include three areas of concern: the question sentence, the correct answer and the distractors. The number of distractors is a parameter and can be determined by the user. The application will generate a question, the correct answer, and the specified number of incorrect answers. The question sentence can be a fill-in-the-blanks or a Wh-question, so these types of questions belong to the category of multiple-choice questions.

### 3.2 True/False Questions

True/False questions are also known as polar or general questions [21]. They have only two possible answers, either "True" or "False". However, it could also be "Yes" or "No", "Agree" or "Disagree", or any other suitable pair of mutually exclusive responses. One of the huge drawbacks of the True/False questions is that the learner has a 50% chance of choosing the correct answer, which could be inadequate for testing the actual knowledge. However, many educational institutions and organizations use these types of questions during assessments. They can ask tricky questions and confuse the learner so that they can test his or her understanding.

### 3.3 Fill-in-the-blanks Questions

A fill-in-the-blanks question consists of a sentence with a blank space where the student can fill the missing word. This type of question can also be clubbed with multiple choice and is easy to evaluate automatically. One famous test using fill-in-the-blank questions is the cloze test, also called the cloze deletion test or the occlusion test. A cloze test is an exercise, test, or assessment consisting of a portion of language with certain items, words, or signs removed, where the participant is asked to replace the missing language item [33].

In our subsystem of CrsMgr, the different types of questions are mixed, aiming for diverse and captivating questions.

### 3.4 Datasets

To train our models, we need a labelled dataset of passages, possible questions and answers retrieved from the passage. Many datasets such as SQuAD, 30MQA, MS MARCO, RACE, NewsQA, Trivia QA, TabMCQ, SciQ and NarrativeQA contain question-answer sets and are used for training of question-answering machine learning models [23]. These datasets contain millions of rows with data sampled from Wikipedia, web searches or crowd sources. These datasets were mainly collected for reading comprehension tasks, making them a good candidate as our goal is also to evaluate the students' comprehension of the study material.

Other datasets like MCQL are intended for automatic distractors generation [28]. Another well-designed dataset is LearningQ which covers a wide range of learning subjects and contains a large set of document-question pairs and multiple source sentences for question generation [20].

In this subsystem of CrsMgr, the following datasets are employed to train the model: BoolQ [22], SQuAD [30], CoQA [31], MS MARCO [18]. Afterwards, the application uses the text passage that the instructor submits as an input source to generate the quiz questions.

## 4 Challenges Of MCQ Generation

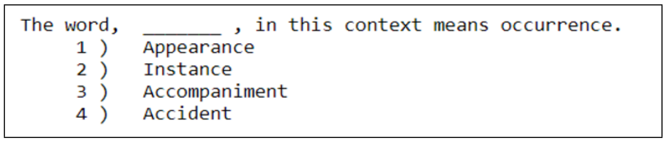
The main challenge is to generate quality questions comparable to the questions a human teacher would create. The contextual language models we have applied significantly improve the results compared to previous rules-based or recurrent-network-based approaches. We found that additional pre-processing steps like summarizing the text, simplifying the sentences, and replacing the pronouns can improve the questions.

Another crucial challenge is the semantic gap. In a natural language, the same meaning can be expressed differently, and the same phrase can have different meanings. The application sometimes fails to identify such words. For example, assume we need to find the distractor for the word "tree" in the context of the computer science domain. However, in NLP it is challenging to find the context of the word when only one word is specified. Therefore, the system might generate irrelevant context distractors. As a result, the student would easily guess the answer from the multiple-choice as the wrong answers will refer to different domains or contexts.

Additional issues are the deviation from the question or unclear questions. The answer deviates from the original question. Sometimes it is difficult for the online assessment to identify the correct keyword to be used to generate the question. If the application generates a vague or ambiguous question, it might lead to multiple answers and thus make it difficult to validate the student's comprehension.

Another problem is the generation of questions that are too simple and too easy for the user. Adding one more layer in the model that classifies the question's difficulty level would help solve such an issue.

In Figure 2, the example question depicts one of the challenges of online question creation. Here the correct answer is 'Instance' and belongs to the computer science domain type of question and answer. CrsMgr uses NLP WordNet to generate distractors (incorrect answers) for the given question, and the generated distractors are not computer science-related words. In turn, NLP WordNet took the word 'Instance' and obtained the synonym but did not consider the domain type. The generated incorrect answers deviated from the question topic. Therefore, it becomes easy for the student to guess the answer and difficult to test the actual knowledge.

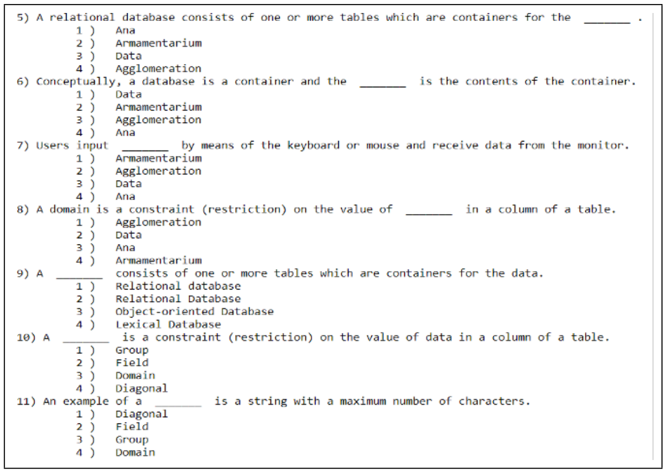


```
The word,  _______ , in this context means occurrence.
        1 )   Appearance
        2 )   Instance
        3 )   Accompaniment
        4 )   Accident
```

**Figure 2.** Question generated by CrsMgr illustrating the challenges of distractor generation.

## 5 Current Status Of CrsMgr-MCQ Generation

In this subsystem of CrsMgr, we generate Multiple Choice Questions (MCQ) from the course material. The instructor has two options either select a passage that will be used to generate the questions or upload the corresponding pdf file. Currently, the system generates fill-in-the-blanks and wh-questions with multiple choice answers as well as True/False questions. In Figure 3. we can see some of the questions that the current subsystem of CrsMgr has generated.



```
5) A relational database consists of one or more tables which are containers for the  _______ .
        1 )   Ana
        2 )   Armamentarium
        3 )   Data
        4 )   Agglomeration
6) Conceptually, a database is a container and the  _______  is the contents of the container.
        1 )   Data
        2 )   Armamentarium
        3 )   Agglomeration
        4 )   Ana
7) Users input  _______  by means of the keyboard or mouse and receive data from the monitor.
        1 )   Armamentarium
        2 )   Agglomeration
        3 )   Data
        4 )   Ana
8) A domain is a constraint (restriction) on the value of  _______  in a column of a table.
        1 )   Agglomeration
        2 )   Data
        3 )   Ana
        4 )   Armamentarium
9) A  _______  consists of one or more tables which are containers for the data.
        1 )   Relational database
        2 )   Relational Database
        3 )   Object-oriented Database
        4 )   Lexical Database
10) A  _______  is a constraint (restriction) on the value of data in a column of a table.
        1 )   Group
        2 )   Field
        3 )   Domain
        4 )   Diagonal
11) An example of a  _______  is a string with a maximum number of characters.
        1 )   Diagonal
        2 )   Field
        3 )   Group
        4 )   Domain
```

**Figure 3.** Sample questions generated by CrsMgr.

The instructor can add, delete or modify the questions and the multiple-choice options and then save the result to the database, after which the quiz becomes available to students for self, non-timed practice.

In order to generate the questions, we use the Summa python library [19] for extractive-based summarization. The library selects the most important sentences in a document or text. We define the summary length as a proportion of the text, and the program considers only mid-length sentences for generating the questions. This pre-processing aims to help the model generate relevant and meaningful questions.

The PDF (Portable Document Format) files are made up of text, vector graphics, raster graphics, and multimedia. The PyMuPDF python library [11] is used to read the PDF file. Unlike HTML, where each text has tags like header or paragraph, PyMuPDF reads the text but does not have tags or structure to identify the passage text. The program extracts font style information like font size and most used fonts sorted by count and colour. The HTML tags are decided based on the count of words of the same font size.

The highest number of words of the same font size is assigned paragraph text. The program considers the sentences marked as paragraph text for the extractive-based summarization. The PDF documents are opened page by page, the HTML tags are identified, and the summary is generated. However, this approach does not give a perfect result. If the PDF page contains any program or table, the program will fail to identify the paragraph text.

In addition to unit testing and integration testing, users were able to test the functionality during the Winter session of 2022 in a real-life scenario. Enrolled students were able to try out the self-study quiz in our web application. The results of the assessments are logged in without resorting to privacy-violating monitoring tools.

# 6 Exploring Transformer Models

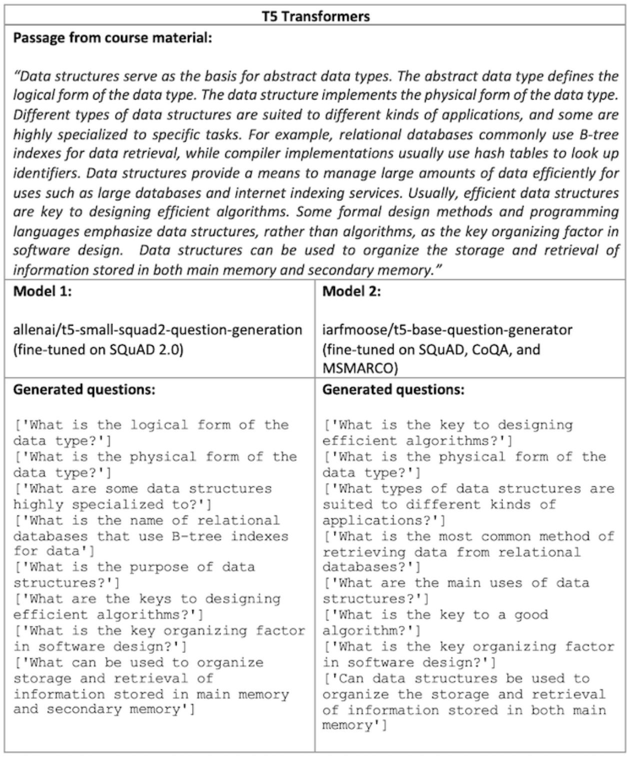
We conducted additional experiments with some Transformer models [6] published by Google as they perform well across a wide variety of NLP tasks. We were interested in how these models can help solve our challenges in generating quizzes from textbooks. These models are worth considering because they work with representations of words within their context. Hence, these models excel at matching phrases that are semantically related although not exact match. This characteristic is helpful for the reading comprehension task.

## 6.1 Methodology

We selected models based on BERT [26] and T5 transformers [29] and fine-tuned on different datasets. These models were trained with the goal to generate reading comprehension-style questions with answers extracted from the source text. The training datasets include a paragraph, corresponding questions, and answers. The purpose of this data format is to fine-tune the models to generate questions and predict answers. We have considered different models. Some models only require the input paragraph as and can generate questions or question-answer pairs. Other models expect both the paragraph and the answer and can output the question. These models can be used in combination with other methods to identify keywords and then ask questions about those keywords. There is a third approach where we ask the model to predict the answer by a given paragraph and question. Detailed results of our experiments can be found in our notebook [7].

## 6.2 Experiment: The role of the training dataset

This experiment aims to verify if the training dataset plays a role in the quality of the questions that can be generated. We have selected four models based on T5 transformers architecture and fine-tuned on different datasets. As we can see in Figure 4, the style of the generated questions depends on the dataset on which the model has been fine-tuned. Some models have the capacity to generate both factual and yes/no questions. This is the case of Model 2, which has been trained on several datasets.



| T5 Transformers | |
|---|---|
| **Passage from course material:**<br>*"Data structures serve as the basis for abstract data types. The abstract data type defines the logical form of the data type. The data structure implements the physical form of the data type. Different types of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks. For example, relational databases commonly use B-tree indexes for data retrieval, while compiler implementations usually use hash tables to look up identifiers. Data structures provide a means to manage large amounts of data efficiently for uses such as large databases and internet indexing services. Usually, efficient data structures are key to designing efficient algorithms. Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design. Data structures can be used to organize the storage and retrieval of information stored in both main memory and secondary memory."* | |
| **Model 1:**<br>allenai/t5-small-squad2-question-generation (fine-tuned on SQuAD 2.0) | **Model 2:**<br>iarfmoose/t5-base-question-generator (fine-tuned on SQuAD, CoQA, and MSMARCO) |
| **Generated questions:**<br>['What is the logical form of the data type?']<br>['What is the physical form of the data type?']<br>['What are some data structures highly specialized to?']<br>['What is the name of relational databases that use B-tree indexes for data']<br>['What is the purpose of data structures?']<br>['What are the keys to designing efficient algorithms?']<br>['What is the key organizing factor in software design?']<br>['What can be used to organize storage and retrieval of information stored in main memory and secondary memory'] | **Generated questions:**<br>['What is the key to designing efficient algorithms?']<br>['What is the physical form of the data type?']<br>['What types of data structures are suited to different kinds of applications?']<br>['What is the most common method of retrieving data from relational databases?']<br>['What are the main uses of data structures?']<br>['What is the key to a good algorithm?']<br>['What is the key organizing factor in software design?']<br>['Can data structures be used to organize the storage and retrieval of information stored in both main memory'] |

**Figure 4.** Questions generated by two T5-based models fine-tuned on different datasets.

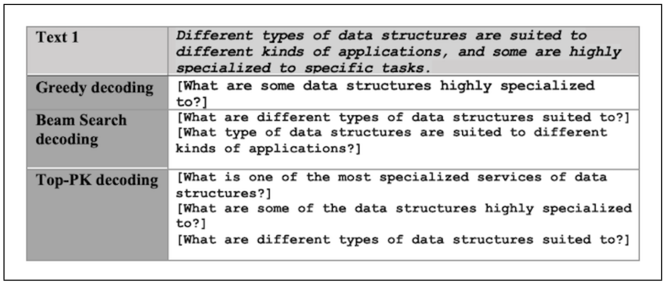
## 6.3 Experiment: The role of the decoder

The purpose of the second experiment was to observe how the decoding method can help us improve the quality and the variety of the questions. In Figure 5, we show the questions generated by the same model but using different decoding methods.

We have tried the following decoding methods:

- Greedy search
- Beam search
- Top-K sampling
- Top-P sampling

We have observed that the decoding strategy can influence the style of the generated questions as it will decide which tokens are selected as output. For example, if we use greedy decoding, we can get stuck in the same sets of words because the greedy algorithm always selects the tokens with the highest probability. On the other hand, beam search will introduce more variety in the output, as it keeps the most likely hypotheses at each step and eventually chooses the hypothesis with the overall highest probability. This way, we include some word sequences in the output that the greedy search might be missing. In Top-K sampling, the K most likely next words are filtered and depending on the distribution and the value of K we can end up selecting very unlikely words. In Top-P sampling (or nucleus sampling), instead of sampling only from the most likely K words, we choose from the minimum number of words whose cumulative probability exceeds the probability P. In conclusion, the different decoding methods introduce more variety in the output questions because human language does not always select the highest probability words as in the greedy search. Based on our observations, we would recommend the beam search decoding because it introduces some variety and, at the same time it, produces more accurate results than the Top-PK strategy.



| Text 1 | *Different types of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks.* |
|---|---|
| Greedy decoding | [What are some data structures highly specialized to?] |
| Beam Search decoding | [What are different types of data structures suited to?]<br>[What type of data structures are suited to different kinds of applications?] |
| Top-PK decoding | [What is one of the most specialized services of data structures?]<br>[What are some of the data structures highly specialized to?]<br>[What are different types of data structures suited to?] |

**Figure 5.** Questions generated by the same model using different decoding methods.

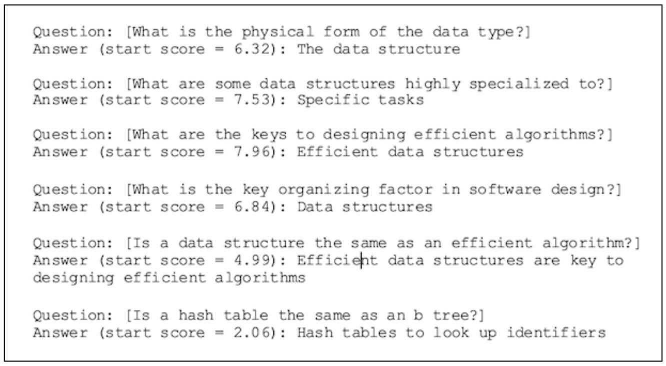
### 6.4 Experiment: BERT for question answering

We have applied the model BertForQuestionAnswering, which is based on BERT architecture [26] and fine-tuned on SQuAD dataset [30]. We have asked the model to generate an answer on three different types of text. One paragraph was selected from a textbook, another paragraph was selected from a video transcript of a lecture, and the third paragraph was selected from a Wikipedia page. We observed that the model worked very well on the Wikipedia paragraph; in second place came the selected paragraph in a textbook, and the performance was worse in the case of video transcription. This observation brings us back to the problem of choosing the appropriate training dataset to help the model learn about the specific domain. Most models are trained on Wikipedia text; however, they have difficulties generalizing on more complex text found in textbooks or unstructured text like video transcripts.

## 7 Results

In summary, we have created a sub-system of CrsMgr able to generate MCQ quizzes from selected course materials. We have encountered several challenges and explored how different language models could help us in the quiz generation task. We focused on models based on GPT-2, BERT and T5 text-to-text architectures. We did not have time to complete our studies of other models like GPT-3, which could further improve the quality of the question-answers sets. Nonetheless, based on our findings, we understand the possible factors that could help us build a model capable of generating quality quizzes. These factors are:

1. Select appropriate training datasets related to the coursework material to fine-tune the models on the specific task of quiz generation in the educational context.
2. Apply beam search as a decoding technique.
3. Assess and eliminate low-quality questions-answers pairs.
4. Pre-process the paragraph appropriately to facilitate the quiz generation.
5. Find better methods to generate distractors.



```
Question: [What is the physical form of the data type?]
Answer (start score = 6.32): The data structure

Question: [What are some data structures highly specialized to?]
Answer (start score = 7.53): Specific tasks

Question: [What are the keys to designing efficient algorithms?]
Answer (start score = 7.96): Efficient data structures

Question: [What is the key organizing factor in software design?]
Answer (start score = 6.84): Data structures

Question: [Is a data structure the same as an efficient algorithm?]
Answer (start score = 4.99): Efficient data structures are key to
designing efficient algorithms

Question: [Is a hash table the same as an b tree?]
Answer (start score = 2.06): Hash tables to look up identifiers
```

**Figure 6.** Selected answers generated by BERT model.

We observed that each model has its advantages and disadvantages. Hence, we need to combine different models to obtain a wider variety of questions and more accurate answers. Another issue is that the training datasets are mostly sourced from Wikipedia, but in the educational context, the course materials could be much more complex. For this reason, we may try other datasets more suited for generating multiple-choice questions in educational context.

## 8 Conclusions and Future Work

This paper describes a work in progress of building a subsystem of CrsMgr for quiz generation. We have tried to adapt some of the existing question-answering and question generation language models, and the results are promising.

However, there is scope for further improvement in the quality of the questions and the answers. Although the pre-trained machine learning models are abundant, they are not specialized in generating multichoice quizzes in an educational context. Further work is needed to determine which models or combinations of models are best for this task.

That is why we propose the following areas for future work:

**Dataset**: Our approach is data-driven; however, the datasets we have used so far are open-sourced large-scale Question-Answer datasets, extracted mainly from Wikipedia and annotated by humans. However, when dealing with specific professional domains, the corpus data is more complex than Wikipedia articles. Therefore, we can further fine-tune our models on domain-specific datasets to improve their relevance and accuracy.

In addition, we can apply different pre-processing techniques to facilitate the work of the model. So far, we have used the summarization technique so that the model can focus on the essential parts of the text. We can apply other data augmentation techniques like named entity resolution, replacing pronouns, noun phrases and verb phrases, simplifying the sentences, paraphrasing and synonym replacement. The goal is to select informative sentences and words to generate the question-answer pairs. We can also better handle unknown words and specific terminology by providing the model with a domain-specific dictionary.

**Model**: Our goal is to generate diverse and relevant question-answers pairs. We plan to explore different encoding and decoding techniques to improve the variety of our quizzes. So far, we have confirmed that the beam search decoding introduces more variety in the generated questions than greedy decoding. It selects several highly probable words depending on the beam size in contrast to the greedy search, which selects only one word - the highest probability word. We can explore other decoding techniques like Monte Carlo Tree Search or Value Guided Beam Search. These decoding methods might introduce more variety in the output questions closer to human language.

Another challenge we face is to ensure the questions, the corresponding answer and the distractors are semantically consistent. We could apply techniques like nearest neighbours or distance measurements to ensure the semantic similarity of the generated distractors.

**Evaluation**: In some cases, our models may not give good results. We need to use the appropriate evaluation metrics. For example, question similarity calculation could exclude redundant questions.

**Further experiments**: We have explored several pre-trained language models; however, there are many more that could become a good base candidate. We plan to try other models and perform hyper-parameter tuning to select the best models. We can combine the best ones in an ensemble to improve the outcome.

Finally, the automated generation of MCQ quizzes is challenging and extremely broad while having important practical applications in reality. Our system's final version will have different types of questions to make self-learning enjoyable and valuable. Learners can use this system as a mock test when preparing for final examinations, and teachers can use it to save time and provide more options for their students. Overall, the teaching process will be more worthwhile and efficient.

## References

[1] 2022. *Class Maker - Quiz Maker for Business and Education*. Retrieved July 24, 2022 from https://www.classmarker.com/

[2] 2022. *Edmodo*. Retrieved July 24, 2022 from https://new.edmodo.com/

[3] 2022. *ExamTime Quizzes*. Retrieved July 24, 2022 from https://examtimequiz.com/

[4] 2022. *Gnowledge*. Retrieved July 24, 2022 from https://www.gnowledge.com/

[5] 2022. *GoToQuiz*. Retrieved July 24, 2022 from https://www.gotoquiz.com/

[6] 2022. *Hugging Face documentation*. Retrieved July 24, 2022 from https://huggingface.co/transformers/usage.html

[7] 2022. *Jupiter notebook link*. Retrieved July 24, 2022 from https://colab.research.google.com/drive/1ULmYhsCGRfLfi6wf9IsxD2jKNCUfwsGs

[8] 2022. *Mentimeter*. Retrieved July 24, 2022 from https://www.mentimeter.com/

[9] 2022. *Online Quiz Creator*. Retrieved July 24, 2022 from https://www.onlinequizcreator.com/

[10] 2022. *ProProfs Quiz Maker*. Retrieved July 24, 2022 from https://www.proprofs.com/quiz-school/

[11] 2022. *PyMuPDF*. Retrieved July 24, 2022 from https://github.com/pymupdf/PyMuPDF

[12] 2022. *QuizStar*. Retrieved July 24, 2022 from http://quizstar.4teachers.org/

[13] 2022. *Socrative Student Response System*. Retrieved July 24, 2022 from https://www.socrative.com/

[14] 2022. *Survey Anyplace - Pointerpro - Mobile Quizzes and Surveys*. Retrieved July 24, 2022 from https://surveyanyplace.com/

[15] 2022. *Testmoz*. Retrieved July 24, 2022 from https://testmoz.com/

[16] 2022. *ThatQuiz*. Retrieved July 24, 2022 from https://www.thatquiz.org/

[17] The Alcade. May 1973. Emmis Communications. ISSN 1535-993X. "Ben D. Wood, 1917, New York, the creator of the Multiple Choice test, has many educational firsts to his credit.".

[18] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[19] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the Similarity Function of TextRank for Automated Summarization. *CoRR* abs/1602.03606 (2016). arXiv:1602.03606 http://arxiv.org/abs/1602.03606

[20] Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. LearningQ: A Large-Scale Dataset for Educational Question Generation. *Proceedings of the International AAAI Conference on Web and Social Media* 12, 1 (Jun. 2018). https://ojs.aaai.org/index.php/ICWSM/article/view/14987

[21] William Chisholm, Louis Tonko Milic, and John A. C. Greppin. 1982. *Interrogativity : a colloquium on the grammar, typology, and pragmatics of questions in seven diverse languages.*

[22] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044* (2019).

[23] Bidyut Das, Mukta Majumder, Santanu Phadikar, and Arif Ahmed Sekh. 2021. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning* 16 (2021), 1–15.

[24] Bipin C. Desai. 1990. *An Introduction to Database Systems.* West, St. Paul, MN. https://spectrum.library.concordia.ca/id/eprint/988586/

[25] Bipin C. Desai and Arlin L Kipling. 2020. *Database Web Programming.* BytePress, Canada. https://spectrum.library.concordia.ca/id/eprint/988529/

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[27] Christiane Fellbaum. 1998. WordNet: An Electronic Lexical Database, chapter A semantic network of English verbs.

[28] Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C. Lee Giles. 2018. Distractor Generation for Multiple Choice Questions Using Learning to Rank. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications.* Association for Computational Linguistics, New Orleans, Louisiana, 284–290. https://doi.org/10.18653/v1/W18-0533

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 140 (2020), 1–67.

[30] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[31] Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics* 7 (2019), 249–266.

[32] Faculty Focus |Higher Ed Teaching & Learning. 2022. *Multiple-Choice Tests: Revisiting the Pros and Con.* Retrieved July 24, 2022 from https://www.facultyfocus.com/articles/educational-assessment/multiple-choice-tests-pros-cons/

[33] Wikipedia contributors. [n.d.]. *Cloze test — Wikipedia, The Free Encyclopedia.* Retrieved July 24, 2022 from https://en.wikipedia.org/w/index.php?title=Cloze_test&oldid=1095203708

[34] Jianhui Zhu. 2016. *Secure CrsMgr: a course manager system.* Master's thesis. Concordia University. https://spectrum.library.concordia.ca/id/eprint/981879/

# Running Temporal Logical Queries on the Relational Model

Samuel Appleby
s.appleby3@newcastle.ac.uk
Newcastle University
School of Computing
United Kingdom

Giacomo Bergami
ngb113@newcastle.ac.uk
Newcastle University
School of Computing
United Kingdom

Graham Morgan
graham.morgan@newcastle.ac.uk
Newcastle University
School of Computing
United Kingdom

## ABSTRACT

State of the art for model checking exploit computationally intensive solutions, bottlenecked by either repeated data access or suboptimal algorithmic implementations. Our solution outperforms the previous solutions while proposing novel temporal logic operators for accessing relational tables.

## CCS CONCEPTS

• **Information systems** → **Association rules**; **Data mining**; *Data scans*; *Data access methods*; **Query optimization**; **Database query processing**; • **Computing methodologies** → *Temporal reasoning*.

## KEYWORDS

Logical Artificial Intelligence, Knowledge Bases, Query Plan, Temporal Logic

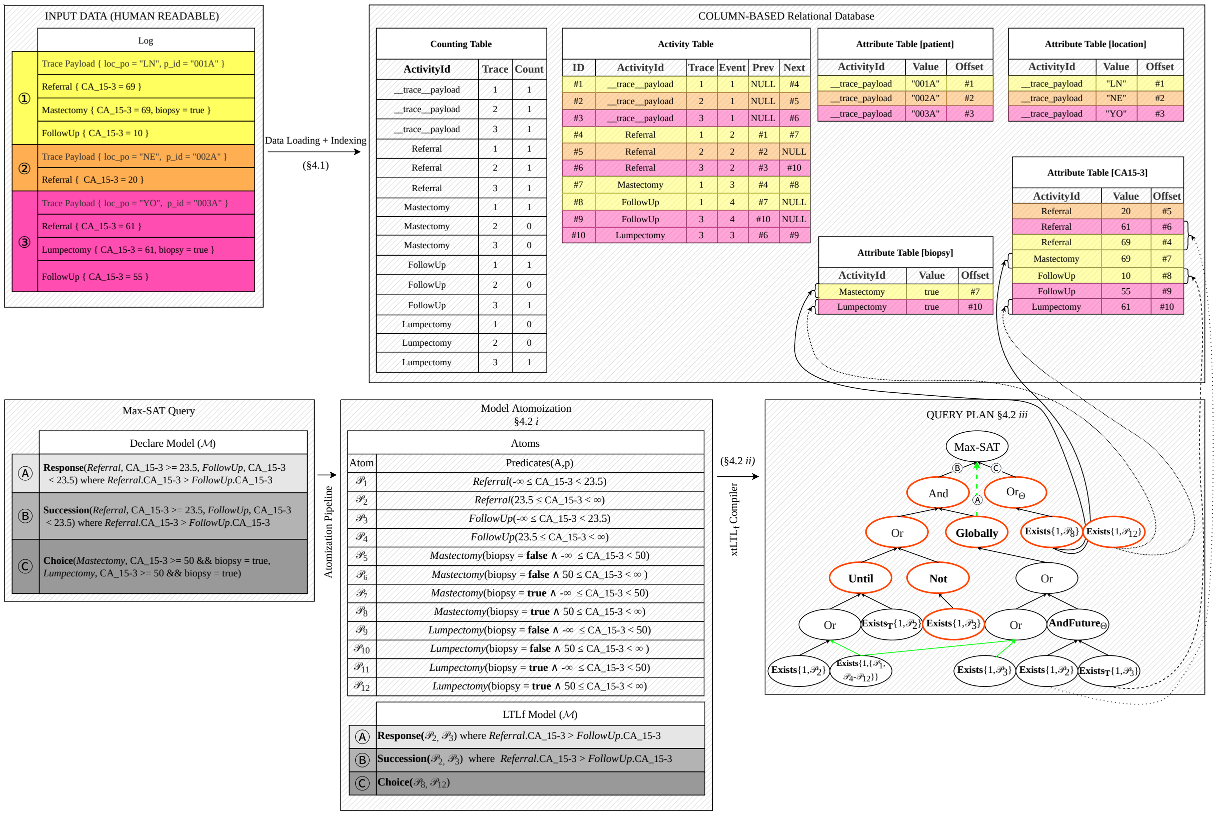
**ACM Reference Format:**
Samuel Appleby, Giacomo Bergami, and Graham Morgan. 2022. Running Temporal Logical Queries on the Relational Model. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3548785.3548786

## 1 INTRODUCTION

*Conformance checking* is an integral part of Artificial Intelligence bridging data mining and business process management [7]. It assesses whether a sequence of distinguishable events (i.e., a *trace*) conforms to the expected process behaviour represented as a *(process) model* [22]. Each event is associated with both an *activity label* describing the captured event, as well as payload data, either associated to the whole trace or to a specific event. When multiple distinct traces are considered in a log, model checking lists the traces satisfying the model [8]. Non-conforming traces are usually referred to as *deviant* [7]. *Declarative* models are composed of multiple human-readable *clauses* that should be jointly satisfied (i.e., *conjunctive query*) [14]; each of these is the instantiation of a specific behavioural pattern (i.e., *template*) expressing temporal correlations between actions being carried out thus linking preconditions to expected outcomes. Such correlations might also involve $\Theta$-joins between activated and targeted events. Models can be expressed as Finite State Machines [2, 15] but, by doing so, each state will represent a possible state configuration the system might find itself in, for which we need to describe all the reasonable actions and data conditions. This makes graph data-aware model checking as [7] rather inefficient, as the size of these graphs becomes exponential with respect to the original size of the declarative model. As a result, this increases the computational time required for conformance checking. Such models are also incapable of expressing $\Theta$-correlation conditions on the data payload, thus limiting the models' expressiveness.

Conformance checking with declarative models is a well-studied technology at the core of AI's temporal decision making. *Firstly*, conformance checking is adopted when mining a model from logs either containing only positive (or negative) traces [21], or on logs containing both, but where positive traces can be discriminated from the negative ones via behavioural or data conditions, thus allowing to generate both a positive and a negative model [6]. The example proposed in this paper (Figure 1), contains cancer patient records obtained from a hospital (this data is included in our datasets[6]). In healthcare, individuals likely to suffer from an illness should receive treatment, and those that are not suffering should not. Therefore, cases where sufferers not receiving treatment (false negatives) and non-sufferers receiving treatment (false positives) needs to be minimised. Figure 1 proposes a simplified scenario for defining this scenario. We conisder 2 event payload labels: CA 15-3 (cancer antigen concentration in a patients blood), and biopsy (biopsies should be taken before any procedure is acted upon). Our model targets only breast cancer patients with successful therapies, that describes a medical protocol and the desired patients' health condition at each step. Ⓒ states two possible surgical operations for breast tumours are mastectomy or lumpectomy if the biopsy is positive and the CA-13.5 is way above ($\geq$ 50) the guard level being 23.5 units per *ml*, and Ⓐ-Ⓑ any successful treatment should decrease the CA-13.5 levels, which should be below the guard level; such correlation data condition is expressed via a $\Theta$ condition (also indicated as where). A twinned negative model (not in Figure) might better discriminate healthy patients from patients where the therapy was unsuccessful. *Secondly*, conformance checking can also exploit such models for predicting which novel clinical situations represented as traces are likely to adhere to the expected clinical standards. Novel situations can be represented as a log: e.g., in Figure 1, we have three patients: ① a cancer patient with a successful mastectomy, ② a healthy patient, and ③ an unsuccessful lumpectomy, thus suggesting that the patient might still have some cancerous cells. Given the aforementioned model, patient ① will satisfy the model as the surgical operation was successful, ② will not satisfy the model because neither mastectomy nor lumpectomy

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548786



INPUT DATA (HUMAN READABLE)

| | Log |
|---|---|
| ① | Trace Payload { loc_po = "LN", p_id = "001A" } |
| | Referral { CA_15-3 = 69 } |
| | Mastectomy { CA_15-3 = 69, biopsy = true } |
| | FollowUp { CA_15-3 = 10 } |
| ② | Trace Payload { loc_po = "NE", p_id = "002A" } |
| | Referral { CA_15-3 = 20 } |
| ③ | Trace Payload { loc_po = "YO", p_id = "003A" } |
| | Referral { CA_15-3 = 61 } |
| | Lumpectomy { CA_15-3 = 61, biopsy = true } |
| | FollowUp { CA_15-3 = 55 } |

Data Loading + Indexing (§4.1)

COLUMN-BASED Relational Database

**Counting Table**

| ActivityId | Trace | Count |
|---|---|---|
| __trace__payload | 1 | 1 |
| __trace__payload | 2 | 1 |
| __trace__payload | 3 | 1 |
| Referral | 1 | 1 |
| Referral | 2 | 1 |
| Referral | 3 | 1 |
| Mastectomy | 1 | 1 |
| Mastectomy | 2 | 0 |
| Mastectomy | 3 | 0 |
| FollowUp | 1 | 1 |
| FollowUp | 2 | 0 |
| FollowUp | 3 | 1 |
| Lumpectomy | 1 | 0 |
| Lumpectomy | 2 | 0 |
| Lumpectomy | 3 | 1 |

**Activity Table**

| ID | ActivityId | Trace | Event | Prev | Next |
|---|---|---|---|---|---|
| #1 | __trace__payload | 1 | 1 | NULL | #4 |
| #2 | __trace__payload | 2 | 1 | NULL | #5 |
| #3 | __trace__payload | 3 | 1 | NULL | #6 |
| #4 | Referral | 1 | 2 | #1 | #7 |
| #5 | Referral | 2 | 2 | #2 | NULL |
| #6 | Referral | 3 | 2 | #3 | #10 |
| #7 | Mastectomy | 1 | 3 | #4 | #8 |
| #8 | FollowUp | 1 | 4 | #7 | NULL |
| #9 | FollowUp | 3 | 4 | #10 | NULL |
| #10 | Lumpectomy | 3 | 3 | #6 | #9 |

**Attribute Table [patient]**

| ActivityId | Value | Offset |
|---|---|---|
| __trace_payload | "001A" | #1 |
| __trace_payload | "002A" | #2 |
| __trace_payload | "003A" | #3 |

**Attribute Table [location]**

| ActivityId | Value | Offset |
|---|---|---|
| __trace_payload | "LN" | #1 |
| __trace_payload | "NE" | #2 |
| __trace_payload | "YO" | #3 |

**Attribute Table [biopsy]**

| ActivityId | Value | Offset |
|---|---|---|
| Mastectomy | true | #7 |
| Lumpectomy | true | #10 |

**Attribute Table [CA15-3]**

| ActivityId | Value | Offset |
|---|---|---|
| Referral | 20 | #5 |
| Referral | 61 | #6 |
| Referral | 69 | #4 |
| Mastectomy | 69 | #7 |
| FollowUp | 10 | #8 |
| FollowUp | 55 | #9 |
| Lumpectomy | 61 | #10 |

Max-SAT Query

| | Declare Model ($\mathcal{M}$) |
|---|---|
| Ⓐ | **Response**(*Referral*, CA_15-3 >= 23.5, *FollowUp*, CA_15-3 < 23.5) where *Referral*.CA_15-3 > *FollowUp*.CA_15-3 |
| Ⓑ | **Succession**(*Referral*, CA_15-3 >= 23.5, *FollowUp*, CA_15-3 < 23.5) where *Referral*.CA_15-3 > *FollowUp*.CA_15-3 |
| Ⓒ | **Choice**(*Mastectomy*, CA_15-3 >= 50 && biopsy = true, *Lumpectomy*, CA_15-3 >= 50 && biopsy = true) |

Atomization Pipeline

Model Atomoization
§4.2 *i*

| Atom | Predicates(A,p) |
|---|---|
| $\mathscr{P}_1$ | *Referral*($-\infty \leq$ CA_15-3 $< 23.5$) |
| $\mathscr{P}_2$ | *Referral*($23.5 \leq$ CA_15-3 $< \infty$) |
| $\mathscr{P}_3$ | *FollowUp*($-\infty \leq$ CA_15-3 $< 23.5$) |
| $\mathscr{P}_4$ | *FollowUp*($23.5 \leq$ CA_15-3 $< \infty$) |
| $\mathscr{P}_5$ | *Mastectomy*(biopsy = **false** ∧ $-\infty \leq$ CA_15-3 $< 50$) |
| $\mathscr{P}_6$ | *Mastectomy*(biopsy = **false** ∧ $50 \leq$ CA_15-3 $< \infty$ ) |
| $\mathscr{P}_7$ | *Mastectomy*(biopsy = **true** ∧ $-\infty \leq$ CA_15-3 $< 50$) |
| $\mathscr{P}_8$ | *Mastectomy*(biopsy = **true** ∧ $50 \leq$ CA_15-3 $< \infty$) |
| $\mathscr{P}_9$ | *Lumpectomy*(biopsy = **false** ∧ $-\infty \leq$ CA_15-3 $< 50$) |
| $\mathscr{P}_{10}$ | *Lumpectomy*(biopsy = **false** ∧ $50 \leq$ CA_15-3 $< \infty$ ) |
| $\mathscr{P}_{11}$ | *Lumpectomy*(biopsy = **true** ∧ $-\infty \leq$ CA_15-3 $< 50$) |
| $\mathscr{P}_{12}$ | *Lumpectomy*(biopsy = **true** ∧ $50 \leq$ CA_15-3 $< \infty$) |

| | LTLf Model ($\mathcal{M}$) |
|---|---|
| Ⓐ | **Response(**$\mathscr{P}_2, \mathscr{P}_3$) where *Referral*.CA_15-3 > *FollowUp*.CA_15-3 |
| Ⓑ | **Succession(**$\mathscr{P}_2, \mathscr{P}_3$) where *Referral*.CA_15-3 > *FollowUp*.CA_15-3 |
| Ⓒ | **Choice(**$\mathscr{P}_8, \mathscr{P}_{12}$) |

(§4.2 *ii*) xtLTL$_f$ Compiler

QUERY PLAN §4.2 iii
Max-SAT
Ⓑ
Ⓒ
Ⓐ
And
Or$_\Theta$
Or
Globally
Exists{1,$\mathscr{P}_8$}
Exists{1,$\mathscr{P}_{12}$}
Until
Not
Or
Or
Exists$_T${1,$\mathscr{P}_2$}
Exists{1,$\mathscr{P}_3$}
Or
AndFuture$_\Theta$
Exists{1,$\mathscr{P}_2$}
Exists{1,{$\mathscr{P}_1$, $\mathscr{P}_4$-$\mathscr{P}_{12}$}}
Exists{1,$\mathscr{P}_3$}
Exists{1,$\mathscr{P}_2$}
Exists$_T${1,$\mathscr{P}_3$}

**Figure 1: KnoBAB Architecture for Breast Cancer patients. Each trace ①-③ represents one single patient's clinical history, are represented with unique colouring. Please observe that the atomization process does not consider data distribution but rather partitions the data space as described by the data activation and target conditions. In the query plan, green arrows remark access to shared sub-queries as in [4], and thick red ellipses remark which operators are untimed.**

was required ($\mathcal{M}$ is only fulfilled for *successful* procedures), and ③ will not satisfy the target condition, even though the correlation condition was met. Our model of interest should only return ① as an outcome of the conformance checking process.

Real business use case scenarios usually require Θ-correlations. In a goods brokerage scenario [17], items are traded between producers (vendors) and retailers (customers): each transaction starts with a vendor sending a sales quotation to a customer. If an offer is accepted and the order is confirmed, then the item is scheduled for delivery. When ready, a logistic operator collects it. In this scenario, deviant traces either do not reflect the company's rules or will potentially lead to retailers' complaints: e.g., a late delivery complaint can occur only if the date the product is received is greater than the agreed time to receive it as registered in a previous agreement event. This situation cannot be directly expressed as a temporal pattern, as we also need to test the timestamps associated in the data payload. Conformance checking can be applied to several unexplored non-business domains, such as smart contract verification [10]. Most recent **video games** exploit AI features [13]: existing state of the art exploits automata [16] for modelling Non-Player Character's behaviours. As Declarative models and automata are completely equivalent approaches, developers might exploit the former to compactly represent the latter. Furthermore, as debugging AI in video games is a crucial challenge [12], conformance checking solutions might be exploited for debugging unexpected behaviours. As AAA video games already track and log both players and NPC actions[1], it might be also possible to use game logs for distinguishing winning strategies from losing ones [6]. As a result, analysis of an ongoing trace at runtime might 'suggest' actions beneficial to the player based on the game state.

Given that conformance checking is at the heart of both trace validation and model mining, it is of crucial relevance to optimize such a task. Solutions enabling conformance checking via model mining through SQL queries [23, 24] neither explicitly evaluate the satisfiability for every single trace, nor return the traces that satisfy them, but only associate support and confidence values to each of said clauses for model mining purposes. However, as shown in this paper, these queries can be extended to both evaluate satisfiability per trace and return the set of traces satisfying every single clause, thus adhering to the definition from conformance checking literature (§2). In doing so, we are forced to introduce aggregation and nesting operations, which are not generally efficient. This fact is supported by experimental evidence (§5.1), where we also

[1] https://battlefieldtracker.com/

extend the relational representation of traces from [23, 24] (§4.1). Our specific contribution is then the provision of specific operators (xtLTL$_f$) rewriting existing LTL$_f$ operators for the relational model thus efficiently running conformance checking queries in Declare (§3.2). This is also possible through a query plan solution similar to [4] (§4.3), which proves to be more efficient than any solution relying solely on the SQL language. The Query Plan (Figure 1), utilises our proposed xtLTL$_f$ operators (§3.2), as an extension of the traditional LTL$_f$ operators (§2) that logically define a Declare clause (Table 1). While LTL$_f$ operators provide a formal logical temporal definition, xtLTL$_f$ operators are designed to exploit the benefits that a relational model provides. This includes optimized access to the tables defined in Figure 1. For example, the proposed operator Init, which constrains a log to begin with a specified event, can directly access the CountingTable, and exploit offsets to determine the first event per trace. Traditional LTL$_f$ (used for a relational model) would require an entire log scan.

Even state-of-the-art implementations explicitly engineered to solve the conformance checking problem without relying on a relational representation of traces, are not particularly efficient [8]. This solution, not being able to assemble the previously described LTL$_f$ operators within a query plan, can neither minimize the access operations to the trace data nor minimize the re-computation of sub-expressions that appear frequently in the model as recently proposed by [4]. This claim is also supported by analysing the query plan for more recent approaches where no evidence of query optimization over the query plan is given [9, 20]. Further experimental shreds of evidence support such theoretical claims (§5.2): in the first instance, these show that our solution is already more efficient than the state of the art in the literature by two-three orders of magnitude (hundredths\tenths of a millisecond vs. tens of seconds). Furthermore, by using different Declare models composed of several clauses accessing the same activation and target conditions, except the data correlations, our solution exhibits an increase in running time only when new data is accessed and, otherwise, it preserves a constant running time with fewer temporal fluctuations.

*Contributions.* Our proposed solution is then implemented in KnoBAB[2]: we are synthesising logs derived from a system (be it digital or real) to a column-store knowledge base ad-hoc implemented for conformance checking (§4.1). In this instance, we then generate a conformance checking query plan generated from a declarative model (§4.2), be it positive or negative, so to compute desired properties associated with non-deviant traces (§4.3). As per previous remarks, declarative models represent temporal and data constraints that one would expect to hold as true in the non-deviant traces from the twinned system. As such, one can consider those traces returned by the query associated with the declarative model as correct, and the remainder as deviant. As a temporal representation of the declarative model provides a point-of-relativity in the context of correctness (i.e., time itself may dictate if traces maintain correctness throughout the unfolding of the associated events), the considerations of such temporal issues significantly increase the time spent for checking the meeting of the requirements. Our contributions include: (*i*) an extension of the log representation from [23, 24] with a CountingTable and a column-based relational model for representing data payloads (§4.1, upper part of Figure 1), (*ii*) a query compiler (§4.2) transforming each Declare model into a DAG query plan (lower part of Figure 1), (*iii*) a relational formulation of traditional LTL$_f$ as xtLTL$_f$, and (*iv*) an execution engine running the DAG either sequentially or in parallel (§4.3)

## 2 RELATED WORK

*XES Log Model.* (Data) *payloads* are maps associating attributes (i.e., *keys*) to data values. Given a finite set of activity labels Act, an event $\sigma_j^i$ is a pair $\langle \mathsf{a}, p\rangle$, where $\mathsf{a} \in \mathsf{Act}$ is an activity label, and $p$ is a payload, mapping each key to a single value. A *trace* $\sigma^i$ is a temporally-ordered and finite sequence of distinct events $\sigma^i = \sigma_1^i \cdots \sigma_n^i$, modelling a process run. All events within the same trace associate the same values to the same trace keys. A log $\mathcal{L}$ is a finite set of traces $\left\{ \sigma^1, \dots, \sigma^m \right\}$. We denote $\Sigma \subseteq \mathsf{Act}$ as the set of all the distinct activity labels in the log. If a payload is also associated to the whole trace, then this can be easily mimicked by adding an extra event containing such a payload, __trace_payload, at the beginning of the trace. This is evidenced from Table 2, where the BPIC 2012 dataset contains by default 24 unique event labels, but after injecting the __trace_payload event, this increase to 25. This characterization is compliant with the eXtensible Event Stream (XES) format, which is the *de facto* standard for event logs [1].

*Conformance Checking.* Temporal declarative languages pinpoint recurring temporal patterns in highly variable scenarios so as to describe them compactly for both machines and humans [19]. Every single temporal pattern is expressed through *templates* (i.e., an abstract parameterized property: Table 1 column 2), which are then instantiated on a set of real activation, target, or correlation conditions. We can then categorize each Declare template from [14] by means of these conditions and the ability to express correlations between two temporally distant events happening in one trace: simple templates (Table 1, rows 1-3) only involving activation conditions; (mutual) correlation templates (rows from 4 to 15), which describe a dependency between two activation and target conditions, thus including correlations between the two; and negative relation templates (last 2 rows), which describe a negative dependency between two events in correlation. Despite these templates may appear quite similar, but they generate completely different finite state machines, thus suggesting that these conditions are not interchangeable[3]. Figure 2 exemplifies the behavioural difference between two clauses differing only on the template of choice. As a semantics, Declare adopts Linear Temporal Logic over finite traces (LTL$_f$), which interprets formulae over an unbounded, yet finite linear sequence of states. Given a trace $\sigma^i$, the evaluation of a formula $\varphi$ is done in a given state (i.e., event id, or position) of the trace, and we use the notation $\sigma_j^i \vDash \varphi$ to express that $\varphi$ holds starting from the $j$-th event of the $i$-th trace. We also use $\sigma^i \vDash \varphi$ as a shortcut notation for $\sigma_0^i \vDash \varphi$. This denotes that the entire trace $\sigma^i$ *satisfies* $\varphi$. Given that a Declare Model is composed of a set of clauses $\mathcal{M} = \{ c_l \}_{l \leq n, n \in \mathbb{N}}$ which have to be contemporarily satisfied in order to be true, we say that a trace $\sigma^i$ is *conformant* to a model if such a trace satisfies

[2] https://github.com/datagram-db/knobab

[3] http://ltlf2dfa.diag.uniroma1.it/

**Table 1: Declare templates illustrated as exemplifying clauses.** $A \wedge p$ **(**$B \wedge q$**) represents the *activation* (*target*) condition,** $A$ **(**$B$**) denotes the activity label, and** $p$ **(**$q$**) is the data payload condition.**

| Type | Exemplifying clause ($c_l$) | Natural Language Specification for Traces | LTL$_\text{f}$ Semantics ($[[c_l]]$) |
|---|---|---|---|
| *Simple* | $\mathsf{Init}(A, p)$ | The trace should start with an activation | $A \wedge p$ |
| | $\mathsf{Exists}(A, p, n)$ | Activations should occur at least $n$ times | $\mathbf{F}(A \wedge p \wedge \mathbf{X}([[\mathsf{Exists}(A, p, n-1)]]))$ |
| | $\mathsf{Absence}(A, p, n \quad 1)$ | Activations should occur at most $n$ times | $\neg[[\mathsf{Exists}(A, p, n \quad 1)]]$ |
| | $\mathsf{Precedence}(A, p, B, q)$ | Events preceding the activations should not satisfy the target | $\neg(B \wedge p)\ \mathbf{W}\ (A \wedge p)$ |
| *(Mutual) Correlation* | $\mathsf{ChainPrecedence}(A, p, B, q)$ | The activation is immediately preceded by the target. | $\mathbf{G}(\mathbf{X}(A \wedge p) \Rightarrow (B \wedge q))$ |
| | $\mathsf{Choice}(A, p, A', p')$ | One of the two activation conditions must appear. | $\mathbf{F}(A \wedge p) \vee \mathbf{F}(A' \wedge p')$ |
| | $\mathsf{Response}(A, p, B, q)$ | The activation is either followed by or simultaneous to the target. | $\mathbf{G}((A \wedge p) \Rightarrow \mathbf{F}(B \wedge q))$ |
| | $\mathsf{ChainResponse}(A, p, B, q)$ | The activation is immediately followed by the target. | $\mathbf{G}((A \wedge p) \Rightarrow \mathbf{X}(B \wedge q))$ |
| | $\mathsf{RespExistence}(A, p, B, q)$ | The activation requires the existence of the target. | $\mathbf{F}(A \wedge p) \Rightarrow \mathbf{F}(B \wedge q)$ |
| | $\mathsf{ExlChoice}(A, p, A', p')$ | Only one activation condition must happen. | $[[\mathsf{Choice}(\mathsf{A}, p, \mathsf{A'}, p')]] \wedge [[\mathsf{NotCoExistence}(\mathsf{A}, p, \mathsf{A'}, p')]]$ |
| | $\mathsf{CoExistence}(A, p, B, q)$ | RespExistence, and vice versa. | $[[\mathsf{RespExistence}(\mathsf{A}, p, \mathsf{B}, q)]] \wedge [[\mathsf{RespExistence}(\mathsf{B}, q, \mathsf{A}, p)]]$ |
| | $\mathsf{Succession}(A, p, B, q)$ | The target should only follow the activation. | $[[\mathsf{Precedence}(\mathsf{A}, p, \mathsf{B}, q)]] \wedge [[\mathsf{Response}(\mathsf{A}, p, \mathsf{B}, q)]]$ |
| | $\mathsf{ChainSuccession}(A, p, B, q)$ | Activation immediately follows the target, and the target immediately preceeds the activation. | $\mathbf{G}((A \wedge p) \Leftrightarrow \mathbf{X}(B \wedge q))$ |
| | $\mathsf{AltResponse}(A, p, B, q)$ | If an activation occurs, no other activations must happen until the target occurs. | $\mathbf{G}((A \wedge p) \Rightarrow (\neg(A \wedge p)\ \mathbf{U}\ (B \wedge q)))$ |
| | $\mathsf{AltPrecedence}(A, p, B, q)$ | Every activation must be preceded by an target, without any other activation in between | $[[\mathsf{Precedence}(\mathsf{A}, p, \mathsf{B}, q)]] \wedge \mathbf{G}((A \wedge p) \Rightarrow \mathbf{X}(\neg(A \wedge p)\ \mathbf{W}\ (B \wedge q))$ |
| *Not.* | $\mathsf{NotCoExistence}(A, p, B, q)$ | The activation nand the target happen. | $\neg(\mathbf{F}(A \wedge p) \wedge \mathbf{F}(B \wedge q))$ |
| | $\mathsf{NotSuccession}(A, p, B, q)$ | The activation requires that no target condition should follow. | $\mathbf{G}((A \wedge p) \Rightarrow \neg\mathbf{F}(B \wedge q))$ |

the LTL$_\text{f}$ semantics $[\![c_l]\!]$ associated to each clause[4] $c_l$. Therefore, the Maximum-SATisfiability problem (Max-SAT) for each trace counts the ratio between the satisfied clauses over the whole model size. An LTL$_\text{f}$ formula $\varphi$ is built by extending propositional logic with temporal operators in bold:

$$\varphi := \mathsf{A} \wedge p \mid \neg\varphi \mid \varphi \vee \varphi' \mid \varphi \wedge \varphi' \mid \mathbf{X}\varphi \mid \mathbf{G}\varphi \mid \mathbf{F}\varphi \mid \varphi\ \mathbf{U}\ \varphi'$$

where ne**X**t ($\mathbf{X}\varphi$) denotes that the condition $\varphi$ should occur from the next state, **G**lobally ($\mathbf{G}\varphi$) denotes that the condition has to hold on the entire subsequent path, **F**uture ($\mathbf{F}\varphi$) denotes that the condition should occur somewhere on the subsequent path, and **U**ntil as $\varphi\ \mathbf{U}\ \varphi'$ denotes that $\varphi$ has to hold at least until $\varphi'$ becomes true, either at the current or a future state. Generally, binary operators bridge activation and target conditions appearing in two distinct sub-formulæ. Some operators can be seen as syntactic sugar: **W**eakUntil is denoted as $\varphi\ \mathbf{W}\ \varphi' := \varphi\ \mathbf{U}\ \varphi' \vee \mathbf{G}\varphi$, while the implication can be rewritten as $\varphi \Rightarrow \varphi' := (\neg\varphi) \vee (\varphi \wedge \varphi')$. Similarly to relational algebra, these operators also support equivalence rules, thus allowing to rewrite a given LTL$_\text{f}$ expression in an equivalent one that might be more efficient to compute.

Despite this formulation has been already extended so to support correlation constraints [8], such a solution is affected by the following two deficiencies: first, correlation conditions have to be represented alongside the target condition levels, thus hampering the exploitation of efficient relational database algorithms for correlation conditions via joins. Furthermore, these operators can only assess the validity of one trace at a time while, on the other hand, we might need to assess the satisfiability of multiple traces at the same time by composing partial results returned by every single operator. These operators cannot be directly exploited as query operators, where multiple traces are considered contemporarily. For this reasons, §3.2 proposes a reformulation of such operators.

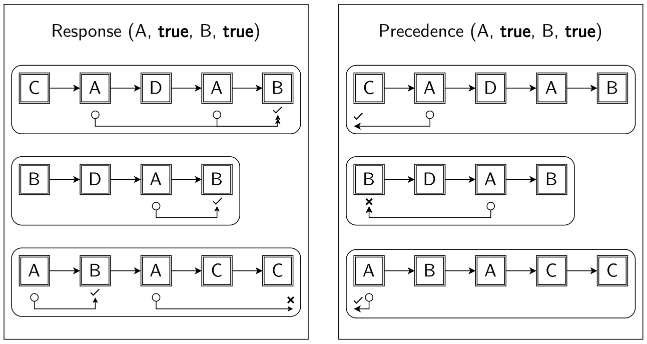

Response (A, **true**, B, **true**)
C
A
D
A
B
B
D
A
B
A
B
A
C
C
Precedence (A, **true**, B, **true**)
C
A
D
A
B
B
D
A
B
A
B
A
C
C

**Figure 2: Traces describing the events generated by each hospital unit: those are temporally ordered events associated to *activity labels* (boxed). Activated (or targeted) events here circled (or ticked/crossed). Ticks (or crosses) indicate a (un)successful match of a target condition.**

*Data-Aware Conformance Checking.* `Declare Analyzer`[5] [8] proposes one of the latest solutions for conformance checking over data-aware logs. Declare templates are decomposed into LTL$_\text{f}$ expressions (as per the last column of Table 1), that not only contain event information, but a payload associated to each event per clause. Such solution does not exploit RDBMS's benefits where query optimisations enhance query running times. So, no possible performance gains by shared sub-queries is considered so to minimize the data access, e.g., by conveniently structuring queries in a query plan [4]. In addition, the authors scan all of the traces completely for each Declare clause, while our proposed solution minimizes the data access by only accessing the data relevant for running the model-checking query. As their solution does not exploit multiple queries running processes, sub-queries or entire clauses appearing

[4] More formally, $\sigma^i \vDash \mathcal{M} \Leftrightarrow \forall c_l \in \mathcal{M}.\sigma^i \vDash [\![c_l]\!]$.

[5] http://www.promtools.org/doku.php?id=prom611

multiple times in the model might be recomputed multiple times, thus tampering with the overall running time. As per their implementation of the LTL$_\mathrm{f}$ operators, authors do not exploit efficient relational algebra operators when possible, as full-outer-theta-joins (or theta-joins) for unions (or conjunctions) with correlation conditions. Last, each clause is completely hardcoded and, as they do not support novel templates via the definition of novel LTL$_\mathrm{f}$ formulae, as we instead do. The addition of further Declare clauses would require an entirely new implementation. KnoBAB, on the other hand, supports the definition of potential new Declare templates via configuration files loaded at warm-up, thus enabling a more general result that goes beyond the Declare language and that can be applied to any temporal specification exploiting LTL$_\mathrm{f}$.

A more recent approach [3] defined specific data structures for a limited support of declarative queries in sublinear-time. Still, this approach has the major shortcoming of pre-computing the possible Precedence or Response queries at loading time. This approach does not scale up for other possible declarative templates, as this might require to extend the data representation with additional data structures. On the other hand, our proposed approach is query independent and supports all of the possible queries that might be expressed in xtLTL$_\mathrm{f}$. Furthermore, this approach supports logs with neither trace nor event payload, thus preventing from easily extending it with activation, target, and correlation conditions involving data predicates. As this approach had a limited query expressive power, it was not considered in our benchmarks.

*Process Mining through Conformance Checking.* Some approaches utilise conformance checking as a mechanism to mine declarative models from an event log: a scoring function tests the validity of each possible clause over each possible trace. SQLMiner [24] does so via SQL queries [23] where each specified declarative template is converted into a SQL query. E.g., given the SQL formulation for the Response template, the query returns a table (Activation,Target,Score) where each row $\langle \mathsf{a}, \mathsf{b}, s\rangle$ represents a candidate clause Response($\mathsf{a}$, **true**, $\mathsf{b}$, **true**), and $s$ is its score.

Each event log, as well as each activation and target activity label for generating the candidate Declare constraints to be tested, are stored in distinct relational tables. While the former are represented in Log(Id,Trace,ActivityId,Event), the latter are stored in Actions(ActivationId,TargetId). The authors consider Support and Confidence scoring functions to determine the precision and reliability of the calculation. Records which do not pass pre-determined Support and Confidence thresholds are filtered out from the data. While SQL also supports data constraints, this solution considers Declare clauses with neither activation, nor target, nor correlation ones with payload predicates. This problem is also shared with more recent approaches where, despite SQL syntax is extended, no evidence of data predicates is given [20].

Despite the authors exploit data perspectives in 'Resource Assignment Constraints' clauses, distinct from the Declare ones, only trace payload conditions are considered. Instead, KnoBAB supports payload information and predicate testing both *per trace* and *per event* (see §2), which could also be stored in a separate table as SQLMiner suggests, thus providing greater expressiveness per clause. SQLMiner queries can be chained together using **`SET UNION`**, though this provides no possibility for testing which are the clauses that are satisfied by the majority of the traces (Max-SAT). These query plans are not optimized as in [4], thus failing at both minimizing the data access and running multiple shared sub-queries only once. This is inferior to KnoBAB, which has the ability to process multiple declarative clauses from disparate templates.

# 3 LOGICAL MODEL

## 3.1 (Intermediate) Result Representation

Within the computation pipeline, (intermediate) results are represented as a set of triplets $\langle i, j, L\rangle$ representing that, starting from event $\sigma^i_j$ in trace $\sigma^i$, we might observe activation, target, or correlation conditions in $L$, an ordered vector. While for activation and target we preserve the matched event id, correlations keep track of both the activation and the target condition leading to the satisfaction of a given $\Theta$ predicate (see the next section). This is a sensible representation, as per declarative constraints, it may exist only one possible $\Theta$ predicate. Such triplets are sorted by trace id and event id, and operators manipulating those (§3.2) guarantee that only one triplet should appear per unique trace and event id. This guarantees efficient join operations across different intermediate results, as well as efficient counting of the satisfied conditions for each trace. E.g., Clause Ⓒ from Figure 1 requires access to just AttributeTables, as all of the activity labels are associated to data conditions. The offset from the attribute tables can then be used to identify the trace and event associated to the data condition (if fulfilled). When we want to return events for which $\mathcal{P}_{12}$ holds, we need to only consider the data associated to `Lumpectomy` events having a positive biopsy and levels of CA15.3 greater than 50. This will require the intersection of the events related to biopsy with the ones related to CA15.3. The selected rows are then converted into the intermediate result representation ad intersected; in this situation, we only obtain $\{\ \langle 3, 3, \{A(3)\}\rangle\ \}$, as the only event meeting such requirements is the third from the third trace. As we are going to see in the next paragraph, A is the container of matched activation conditions. Similarly, $\mathcal{P}_8$ will return $\{\ \langle 1, 3, \{A(3)\}\rangle\ \}$, thus obtaining $\{\ \langle 1, 3, \{A(3)\}, \langle 3, 3, \{A(3)\}\rangle\rangle\ \}$ as a final result associated to Ⓒ: this remarks that only traces ① and ③ describe patients that underwent a surgical operation under such conditions.

Our proposed representation is different from the one provided by [8] which cannot represent for each event within a trace all the possible activation, target, or join condition happening in the future, as it is impossible to represent single trace events that are not necessarily represented by activation or target conditions. As observed in §2, this information is required for checking the satisfiability of $\varphi$ while jointly visiting both the trace (now represented as subsequent rows in the result representation) and the formula. In fact, authors exploit a hash map of hash maps, associating each trace to the collected activation conditions which, in turn, might be associated with further target conditions. This solution is even less efficient than exploiting sorted linear data structures.

## 3.2 eXTended LTL$_\mathrm{f}$ operators

$$\begin{aligned}\phi := \ & \mathsf{Init}_{A/T}(A,p) \mid \mathsf{End}_{A/T}(A,p) \mid \mathsf{Exists}_{A/T}(n,A,p) \mid \mathsf{Absence}_{A/T}(n,A,p)\\ & \mid \mathsf{Next}(\phi) \mid \mathsf{Globally}(\phi) \mid \mathsf{Future}(\phi) \mid \mathsf{Not}(\phi)\\ & \mid \mathsf{Or}(\phi,\phi',\Theta) \mid \mathsf{And}(\phi,\phi',\Theta) \mid \mathsf{Until}(\phi,\phi',\Theta)\\ & \mid \mathsf{AndGlobally}(\phi,\phi',\Theta) \mid \mathsf{AndFuture}(\phi,\phi',\Theta) \mid \mathsf{AndNextGlobally}(\phi,\phi',\Theta)\end{aligned}$$

**Algorithm 1** `xt`LTL$_f$ pseudocode implementation for the basic timed operators

```
function FUTURE($\phi$)
    for all $\langle t, e, L\rangle \in \phi$ do yield $\langle t, e, \bigcup\{ L' \mid \langle t, e', L'\rangle \in \phi$ and $e' \geq e \}\rangle$
    end for
function GLOBALLY($\phi$)
    for all $\langle t, e, L\rangle \in \phi$ do
        $E \leftarrow \{ e' \mid \langle t, e', L'\rangle \in \phi$ and $e' \geq e \}$
        if $|E| = \ell_t - e$ then yield $\langle t, e, \bigcup\{ L' \mid \langle t, e', L'\rangle \in \phi$ and $e' \in E \}\rangle$ end if
    end for
function NEXT($\phi$)
    for all $\langle t, e, L\rangle \in \phi$ s.t. $e > 1$ do yield $\langle t, e-1, L\rangle$
    end for
function COMMONJOIN($\phi, \phi', \Theta, isDisjunctive$)
    $it$ ←Iterator($\phi$), $it'$ ←Iterator($\phi'$)
    while $it \neq \emptyset$ and $it' \neq \emptyset$ do
        $\langle t, e, L\rangle \leftarrow$ current($it$), $\langle t', e', L'\rangle \leftarrow$ current($it'$)
        if $t = t'$ and $e = e'$ then
            $L'' \leftarrow \emptyset$
            if $\Theta \neq$ true and $L \neq \emptyset$ and $L' \neq \emptyset$ then
                for all $A(m) \in L$ and $T(n) \in L'$ s.t. $\Theta(m, n)$ do
                    $L'' \leftarrow L'' \cup \{ M(m, n) \}$
                end for
            else
                if $L = \emptyset$ then $L'' \leftarrow \{A(e)\}$ else $L'' \leftarrow L$
                if $L' = \emptyset$ then $L'' \leftarrow L \cup \{T(e')\}$ else $L'' \leftarrow L'' \cup L'$
            end if
            if $L'' \neq \emptyset$ then yield $\langle t, e, L''\rangle$;
            next($it$); next($it'$);
        else if $t < t'$ or ($t = t'$ and $e < e'$) then
            if $isDisjunctive$ then yield $\langle t, e, L\rangle$ end if
            next($it$)
        else
            if $isDisjunctive$ then yield $\langle t', e', L'\rangle$ end if
            next($it'$)
        end if
    end while
function AND($\phi, \phi', \Theta$) COMMONJOIN($\phi, \phi', \Theta$, false)
function OR($\phi, \phi', \Theta$) COMMONJOIN($\phi, \phi', \Theta$, true)
function UNTIL($\phi, \phi', \Theta$)
    for all $t$ s.t. $\langle t, i', L'\rangle \in \phi'$ do
        $\alpha \leftarrow 1$; $Map \leftarrow \{\}$; $i \leftarrow \min_\iota \langle t, \iota, L\rangle \in \phi'$; $I \leftarrow \max_\iota \langle t, \iota, L_I\rangle \quad 1$
        while $i < I$ do
            if $\alpha = i$ then
                $Map[\alpha] \leftarrow Map[\alpha] \cup L'$
                $i \leftarrow \min_{\iota, \iota > i} \langle t, \iota, L\rangle$
            else if exists $\langle t, j, L_j\rangle \in \phi$ s.t. $j < i$ then
                if $\langle t, \alpha, L_\alpha\rangle, \langle t, \alpha \quad 1, L_{\alpha \quad 1}\rangle, \dots, \langle t, i-1, L_{i-1}\rangle \in \phi$,
                    and $\Theta(i, j)$ for all $T(j) \in L_\alpha \cup \dots \cup L_{i-1}$ then
                    $Map[\alpha] \leftarrow Map[\alpha] \cup \{ M(k, i) \mid T(k) \in L_\alpha \cup \dots \cup L_{i-1} \}$
                    $i \leftarrow \min_{\iota, \iota > i} \langle t, \iota, L\rangle \in \phi'$
                else $\alpha \leftarrow \alpha \quad 1$
                end if
            else $\alpha \leftarrow i$
            end if
        end while
        for all $\langle i, L\rangle \in Map$ do yield $\langle t, i, L\rangle$
        end for
    end for
```

We extended LTL$_f$ operators (`xt`LTL$_f$) directly exploited by our pipeline. Operators in the first line filter traces' events and represent these into the previously-described result representation. Init (End) returns the events at the beginning (end) of each trace satisfying the condition $A \wedge p$. Similarly to [8], each of these operators might be expressed as either an untimed or as a timed specification. Any operator will be considered timed by default when appearing inside a timed operator, like Next, Globally, Future, Until, and any other composed operator from the last line. E.g., In Figure 1, Exists$(1, \mathcal{P}_3)$ is a shorthand for Exists$(1, \text{FollowUp}, -\infty < \text{CA-15.3} < 23.5)$, as each atom always associates an activity label to a payload condition. The operator associated to Absence$(1, \mathcal{P}_3)$ is untimed, while the Exists$(1, \mathcal{P}_3)$ descendant of Globally is timed. While the timed definition returns a tuple $\langle i, j, L\rangle$ for each possible event $\sigma^i_j$ within the trace $\sigma^i$ where the formula holds, the untimed specification only checks whether the formula holds at the beginning of the trace. E.g., untimed Exists (Absence) returns the first event trace if at least $n$ (at most $n-1$) events satisfy $A \wedge p$, while the timed version returns the events satisfying (not satisfying) $A \wedge p$ (always $n = 1$). All of these operators might be optionally marked as returning either an activation ($A$) or a target ($T$) condition, so that each $\langle i, j, L\rangle$ triplet has $L = \{A(j)\}$ or $L = \{T(j)\}$; when no mark is specified, $L$ is empty. To wrap up the previous example, the timed Exists$(1, \mathcal{P}_3)$ will list the events where $\mathcal{P}_3$ happened, $\{ \langle 1, 3, \{A(3)\}\rangle \}$, while the untimed version will just list the traces where such event happened and collect the event of interests in $L$.

The next two lines report the same operators described in §2 with the addition of the explicit correlation conditions over activation and target conditions for each binary operator. Algorithm ?? provides implementations of the timed versions of such operators, due to lack of space untimed versions are not provided, yet available in our codebase: please observe that Next$(\phi)$ keeps unaltered the activation and target conditions from $\phi$ and just returns the events where $\phi$ happens as a subsequent step. Any binary operator supports $\Theta$ conditions: And (and Or) can be expressed as a (full-outer-)$\Theta$-join algorithm over the activation and target conditions stored in $L$ associated with the same event. If at least one activation condition matches one target condition from the same event, those are expressed as a marked correlation condition $M(i, j)$ which is then returned by the join. Regarding the same Choice clause from Figure 1, the correlation condition $\Theta$ associated to Or is then computed for each activation/target match, and if the condition is passed, the resulting match is added to $L$.

The remaining operators merge multiple operators together when a specific implementation outperforms the execution of the operators separately: e.g., AndFuture$(\phi, \phi', \Theta)$ is equivalent to And$(\phi, \text{Future}(\phi'), \Theta)$, but preliminary experiments reveal that the former has a more efficient implementation than computing the latter. This choice was inspired by relational algebra, where $\theta$-joins are usually more efficient than performing a join and a selection operation separately. On the other hand, Implies$(\phi, \phi', \Theta)$ is rewritten as Or(Not$(\phi)$, And$(\phi, \phi', \Theta)$, **true**). As per previous discussion, the left leaf of AndFuture$_\Theta$ in Figure 1 returns all of the referral events with CA 15-3 above the safeguard levels, $\{ \langle 1, 2, \{A(2)\}\rangle, \langle 3, 2, \{A(2)\}\rangle \}$, while the right leaf returns just the follow-up events below such levels, $\{ \langle 1, 4, \{A(4)\}\rangle \}$. The operator AndFuture$_\Theta$ will then return only $\{ \langle 1, 2, \{M(2, 4)\}\rangle \}$, as only the first trace will have a decrease below the safeguard levels from referral to follow-up. Each `xt`LTL$_f$ operator is going to both return and/or accept data in the result representation, thus making such operators closed on such format.

## 4 KNOBAB ARCHITECTURE

The methodology behind its design systematically follows the major architectural components of a relational database, with the only bespoke characteristics of tailoring such solution to the specific problem that we intend to solve (§4.3), that is, computing either the Max-SAT for each log trace, or the Confidence/Support associated to each model clause, or computing the traces satisfying all of the model clauses (*conjunctive query*).

## 4.1 Data Loading

The data loading phase loads logs serialized in multiple formats, thus including the XML-based XES standard, a tab-separated events' activity labels, and the Human Readable Log Format (HRLF) firstly introduced in [7]. We use different data parsers, which are still linked to the same data loading primitives. HRLF also supports the bool data type. This is represented as an integer, where: $val < 1.0 = false, val \geq 1.0 = true$. In Figure 1, booleans are displayed in their traditional way (both in the payload and for activation/target conditions), though this is for visual purposes only E.g., $\text{Exists}(1, \text{Referral}, biopsy = true)$ in our pipeline is $\text{Exists}(1, \text{Referral}, \text{biopsy} \geq 1.0)$.

If the log does not contain data payloads, the entire log can be represented into two relational tables, CountingTable(ActivityId,Trace, Count) and ActivityTable(ActivityId,Trace,Event,Prev,Next). While the former counts the occurrence of each activity label in $\Sigma$ for each trace, the latter lists all of the possible events similarly to SQLMiner. Both tables compactly represent the initial three columns as a 64-bit unsigned integer, which is also used to sort the tables in ascending order. A row $\langle \text{a}, j, h\rangle$ from CountingTable states that there are $h$ events exhibiting the activity label a in the trace $\sigma^j$; each row $\langle \text{a}, j, i, q, q'\rangle$ from ActivityTable states that the $i$-th event of the $j$-th trace ($\sigma_i^j = \langle \text{a}, p\rangle$) is labelled as a, while $q$ (or $q'$) is the pointer to the immediately preceding $\sigma_{i-1}^j$ (or following, $\sigma_{i+1}^j$) event within the trace if any. NULLs from Figure 1 in ActivityTable highlight the start (finish) event of each trace, where there is no possible reference to past (future) events. Trace payload information is injected (as an event) before the first event, which is also contained: all trace payload events contain NULL as Prev.

If, on the other hand, the log is associated to either trace or event payloads, we exploit the query and memory-efficient column-based model [11], thus representing all of the values $v$ associated to a payload key $k$ within the rows from AttributeTable$k$. In our implementation, each row $\langle \text{a}, v, i\rangle$ from AttributeTable$k$(ActivityId,Value,Offset) represents a value $v$ associated to the key $k$, where $i$ determines the location where the event containing the accessed value is located in ActivityTable; this provides the trace id and event id required for the intermediate representation. To perform payload-based queries efficiently, the table is sorted in ascending order by the three columns. As each data condition is always associated with a given activity label, those can be effectively run as data range queries run via binary search algorithms. From Figure 1, all the attributes are stored in distinct tables. Value can contain multiple data types, but each attribute is associated to only one type.

CountingTable is mainly accessed for existential and Exists and Absence templates where no data payload is specified, while ActivityTable is used for either returning all of the events within the log associated to a given activity label or returning all of the events happening at either the beginning or at the end of a trace. Each table AttributeTable$k$, on the other hand, will return all the events satisfying a given condition associated with a specific key $k$.

After loading the whole dataset, the number of the traces within the log $|\mathcal{L}|$, the length $\ell_j$ for each trace $\sigma^j$, and the number of distinct activity labels $|\Sigma|$ is known. Given this, we can get the number of occurrences of each $i$-th activity label from $\Sigma$ in each trace by directly accessing the rows within the CountingTable in Figure 1 are within the range $[|\mathcal{L}| \cdot (i-1)+1,\ |\mathcal{L}| \cdot i]$. The offsets for accessing the *Mastectomy* activity label in CountingTable is $[3 \cdot (4-1)+1, 3 \cdot 4] = [10, 12]$. Given that this counting table computes only for untimed operations, the intermediate result for untimed $\text{Exists}_A(1, \textit{Mastectomy}, \textbf{true})$ is $\{\ \langle 1, 1, \emptyset\rangle\ \}$, as only ① contains such an event. On the other hand, the loading and indexing phase generates an ActivityTable associated with two indices, a primary and a secondary index. While the former returns all of the events associated with a specific activity label, the latter accesses either the first or the last event in a trace. Pointers associated with each record enable traces' temporal scan.

## 4.2 Query Compiler

The query compiler is structured into three main phases. *(i)* The *atomization pipeline* rewrites the data predicates associated with each activity label as a disjunction of mutually exclusive data conditions. We can tune KnoBAB to always atomize each possible activity label if it exists any Declare Constraint associating it to a data condition as in [7], or we can choose to provide such an interval decomposition only to the Declare constraints exhibiting data conditions. While the former approach will maximise the access to the AttributeTables, the latter will maximise the access to the ActTable. By doing so, we can ensure that the data satisfying some given properties can be visited at most once, thus guaranteeing the assumptions from [4] also at the data accessing level. Correlation conditions do not undergo this rewriting step. The atomized model in Figure 1 replaces the non-correlation data predicates with the outcome of the atomization process as in [7].

We *(ii)* rewrite each Declare constraint as a $\texttt{xtLTL}_\text{f}$ formula, where the activations (and the potential target) conditions are instantiated with either just activity labels or also with associated data conditions as per the previous atomization step. Each subexpression appears at most once as in [4] by representing every single node in the query plan at most once: this is ensured by an internal query manager cache. The resulting query plan considering the simultaneous execution of multiple queries can be represented as a Direct Acyclic Graph (DAG). For each declarative clause appearing more than once (e.g., $m > 1$), the associated $\texttt{xtLTL}_\text{f}$ expression will be computed at most once, while its resulting data is going to be accessed $m$ times by the final aggregator: as per Figure 1, despite Response might be considered a subquery of Succession, the Max-SAT is still going to retrieve the output provided by the associated sub-expression. Green arrows remark operators' output shared among operators. Please also observe that operators with the same name and arguments but marked either with activation, target, or no specification are considered different as they provide different results, and therefore are not merged together. This includes distinctions between timed and untimed operators.

Given that our execution engine provides the possibility of running a query plan in either a parallel or a sequential mode, we need an additional step. *(iii)* The previous DAG represents a dependency graph, where a link between an ancestor and one of its descendants implies that the latter has to be computed before the former, thus suggesting an execution order. Figure 1 depicts this as an arrow starting from the ancestor. To enforce that, we perform a lexicographical order over the DAG, through which we compute the maximum depth level associated with each node of the graph. We then represent the query graph as a stack of depth levels, where each operator on it can be run in parallel alongside its siblings. This proves that the computation of Declare Clauses can be reduced into an embarrassingly parallel problem, as the layered execution guarantees that no thread communication needs to happen, and that multiple threads could access contemporary the partial results associated with the immediately-descendant operators, as the former will return all of the events where the condition happened, while the latter will just return the trace event satisfying such condition alongside the required activations/targets listed in $L$. Furthermore, the proposed parallelization ensures minimizing the data access for computing the query. The DAG Figure 1 depicts a query plan.

## 4.3 Execution Engine

At the time of the writing, KnoBAB supports four different types of model aggregation queries: Conjunctive Query, Max-SAT, Confidence, and Support. As we will see at the end of the subsection, these will not require a change on the query plan, but just a different way to integrate the intermediate representation $\phi_i$ returned by each declarative clause $c_i$.

*First*, the execution engine takes both the relational database resulting from the data loading and the DAG returned by the query compiler, and uses the leaf nodes from the latter to access the former. By query plan construction, all of the relevant data parts are going to be accessed at most once and then transformed into the expected intermediate result representation. *Second*, the intermediate results are propagated from the leaves towards each root node associated with a declarative clause $c_k$. Any intermediate representation is always associated with each operator returning it as a temporary primary-memory cache. Each intermediate cache might be completely freed if we are not computing a Confidence query and if the furthest ancestor has already accessed it, or if it is a cache non-associated to an activation required by Confidence and the furthest ancestor has already accessed it. *Third*, when the computation will finish running the shallowest DAG depth level containing the xtLTL$_f$ root associated with the entry-point of each declarative clause $c_k$, each of these operators will have an intermediate result $\phi_k$ stating all the traces satisfying $c_k$.

| Competitor | Dataset | Traces $\lvert\mathcal{L}\rvert$ | Events | Distinct Activities $\lvert\Sigma\rvert$ |
|---|---|---|---|---|
| SQL Miner | BPIC 2011 (original) | 1143 | 150 291 | 624 |
| | BPIC 2011 (10) | 10 | 2613 | 158 |
| | BPIC 2011 (100) | 100 | 12 195 | 276 |
| | BPIC 2011 (1000) | 1000 | 133 935 | 607 |
| Declare Analyzer | BPIC 2012 (original) | 13087 | 262 200 | 24 |

**Table 2: Range of datasets used for benchmarking.**

The Conjunctive Query will return the traces satisfying all of the Declare clauses via the intersection of all of the clauses via And and **true** as a $\Theta$ condition. Max-SAT will count, for each log trace $\sigma_i$, the intermediate results $\phi_k$ associated with each clause $c_k$ containing it, and then provide the ratio of such value over the total number of the model clause $|\mathcal{M}|$. By denoting as $\mathsf{ActLeaves}(\phi_k)$ the untimed union of the intermediate results returned by the activation conditions for the declare clause $c_k \in \mathcal{M}$, the Confidence for $c_k$ is the ratio between the total number of traces returned by $\phi_k$ and the overall traces containing an activation condition. Dividing the total number of traces returned by $\phi_k$ by the total log traces returns the Support. Once each $\phi_k$ per clause $c_k$ is computed, the aggregation functions can be then expressed as follows:

$$\mathsf{ConjQuery}(\phi_1,\dots,\phi_n)=\mathsf{And}(\phi_1,\dots\mathsf{And}(\phi_{n-1},\phi_n,\mathbf{true}),\mathbf{true})$$

$$\text{Max-SAT}(\phi_1,\dots,\phi_n)=\left(\frac{|\{\,k\mid \exists j,L.\,\langle i,j,L\rangle\in\phi_k\,\}|}{|\mathcal{M}|}\right)_{\sigma^i\in\mathcal{L}}$$

$$\textsc{Confidence}(\phi_1,\dots,\phi_n)=\left(\frac{|\{\,i\mid \exists j,L.\,\langle i,j,L\rangle\in\phi_k\,\}|}{|\mathsf{ActLeaves}(\phi_k)|}\right)_{c_k\in\mathcal{M}}$$

$$\textsc{Support}(\phi_1,\dots,\phi_n)=\left(\frac{|\{\,i\mid \exists j,L.\,\langle i,j,L\rangle\in\phi_k\,\}|}{|\mathcal{L}|}\right)_{c_k\in\mathcal{M}}$$

As the user in Figure 1 asks the ratio between satisfied clauses over the model size, the query plan exhibits a Max-SAT aggregation.

# 5 EXPERIMENTAL ANALYSIS

Our benchmarks exploited a Razer Blade Pro on Ubuntu 20.04: Intel Core i7-10875H CPU @ 2.30GHz - 5.10 GHz, 16GB DDR4 2933MHz RAM, 180GB free disk space. Our datasets (Table 2) include 2 real life event logs[6]: BPIC 2011 (Dutch academic hospital log) and BPIC 2012 (Dutch loan company).

## 5.1 SQLMiner

These experiments want to test our working hypotheses for the possibility of engineering a tailored relational database architecture that can outperform process mining through conformance checking running on traditional relational databases. In the latter, no LTL$_f$ operators are exploited but a table similar to `ActivityTable` is exploited. Given that the SQL provided in [23, 24] might only return the Support associated with each candidate Declare clause (`SQLMiner+Support`), we provided the least possible changes to also associate each candidate clause with the set of all the traces satisfying it. This was achieved by both extending the activation condition expressed in SQL and using `array_aggr` included in **PostgreSQL 14.2** to list such traces (`SQLMiner+TraceInfo`). For comparing the same settings in KnoBAB, we run both Max-SAT and Support queries with the difference that, in our case, both of these implementations will always return, per intermediate result specification, the trace information satisfying each possible model clause. For our experiments, we exploited BPIC 2011 dataset from [24]. To test the scalability of the solutions, we recorded the query's runtime with increasing log size: we randomly sampled the log with three sub-logs containing 10, 100 and 1000 traces, while guaranteeing that each sub-log is always a subset of the greater ones. For each sub-log, we generated 8 distinct models as benchmarked

[6] https://dx.doi.org/10.17605/OSF.IO/XWD3V

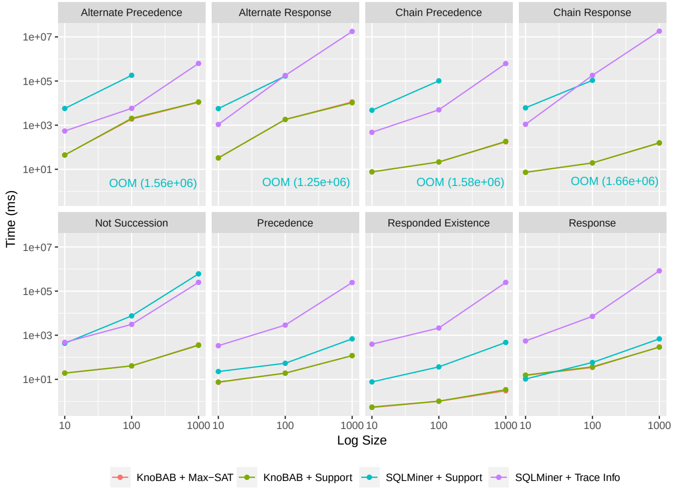


**Figure 3: KnoBAB vs SQLMiner Performance for 25 clauses with frequent activity labels with Support and Trace Information. OOM indicates Out of Secondary Memory for logs containing $10^3$ traces (followed by the time taken for an exception to occur).**

in [23]. Each model consists of 25 clauses instantiating the same Declare template (*elected template*) with different activation and target conditions. Those did not consider payload conditions and were only considering the most frequent activity labels appearing in the sub-log. Models of greater size caused an exponential increase in required secondary memory for SQLMiner (on the order of TB), justifying our approach for a sampled model. In their approach, each model was queried by running the SQL query corresponding to the *elected template*, and the specific activation and target conditions from the model's clauses were distinct rows in the Action table.

The outcome of such experiments is represented in Figure 3, where each plot represents the running time associated with models containing the same *elected template*. In the worst-case scenario (Response), we exhibit similar query running times to SQLMiner. Even so, we are always providing trace information, and in the case of Response, altering the SQL to provide this causes over an order of magnitude increase in complexity. In the best-case scenario, we outperform SQLMiner by at most 5 orders of magnitude. This is because our query plan minimizes the access to the data queries and our computation avoided explicit computations of aggregations. This was achieved by sorting the intermediate results, and, as our operators' implementations guarantee that (intermediate) results are always sorted, counting operations are just linear scans of the intermediate result representation. Our solution never exceeded the 16GB of primary memory while, for some more complex queries (top row of Figure 3), SQLMiner exceeded it, thus proving that our solution is also memory efficient. One of the outstanding examples is RespExistence, where we are greatly more efficient than SQLMiner. This is a clear indicator of the potential gains from utilising our proposed CountTable, summarizing the appearance of activity labels in events per trace by counting their instances. The original SQL query is required to scan the whole Log table (similar to our `ActivityTable`), which contained all of the trace events. We also remind the reader that the CountTable can be efficiently created while scanning the whole log dataset, so no super-linear overhead is added at loading time. This further validates that an adequate tabular representation twinned with xtLTL$_f$ operators extending the LTL$_f$ specification for tabular data provides a suitable solution. Last, the running time of the Max-SAT problem and SUPPORT for KnoBAB exhibited similar running times, while in PostgreSQL those exhibit huge variations depending on the query-plan rewriting performed by the PostgreSQL query engine. For some elected templates, our proposed `SQLMiner+TraceInfo` formulation proved also to be more efficient than the `SQLMiner+Support` queries originally proposed by [23] (which contain no trace information).

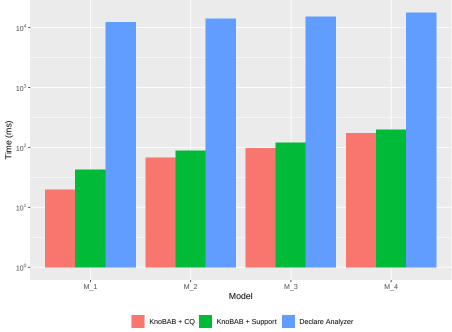


**Figure 4: KnoBAB vs Declare Analyzer Performance.**

## 5.2 Declare Analyzer

The set of experiments on Declare Analyzer have the aim of comparing our proposed solution against a solution tailored for solving Declare Conjunctive Queries over logs running exclusively in primary memory. We chose to exploit via MapDB[7] for log representation, thus making it more similar to a relational database. We exploited the BPIC 2012 dataset, defined in Table 2, also used in [8]. The data was modified so as to efficiently act across the trace payload information. [8] requires the injecting of trace payload information into *each* event. Our implementation, as stated in 2, injects the trace payload as a unique event at the beginning of the trace. The queries (from the same paper) were edited[6], where all the models $M_i$ and $M_j$ with $i < j$ are always the former a subset of the latter, while $M_{i+1}$ increases by 5 from $M_i$. Our experiments indicate that, overall, we are 2-3 times orders of magnitude more performant than DeclareAnalyzer. The conjunctive query denoted as `KnoBAB+CQ` demonstrates greater performance that `KnoBAB+Support`, as the calculations required for the support values per clause are more costly for smaller models. Though this is only within the order of the milliseconds. For an increase in model size, Declare Analyzer has a much greater time increase than KnoBAB (the best case for Declare Analyzer is over an order of magnitude greater than that of KnoBAB). While the linear interpolation of Declare Analyzer provides a slope of $3.47 \cdot 10^2$ *ms* per model size, KnoBAB provides a slope of $10^1$, thus providing an inferior overall growth rate. To explain the abrupt time increase from $M_1$ to $M_2$, we encourage the reader to refer back to the query plan from Figure 1.With each

[7] https://mapdb.org/

increase in model size, entirely new event labels and data payloads are considered (albeit the conditions within each sub-model are similar). As KnoBAB thrives when data access can be limited, the addition of new data requires more decomposition within the atomization pipeline, and, as more atoms are now considered, querying will also suffer as more data is going to be accessed. As a result, the complexity increase is worse than examples tailored to benefit data access limiting as in the previous scenarios where queries were sharing multiple and frequent activity conditions. Still, Declare Analyzer will always completely scan all the events by design despite, for some queries, we might exclude scanning irrelevant trace events.

## 6 CONCLUSIONS AND FUTURE WORKS

We propose KnoBAB, a fully relational database architecture for computing Conformance Checking via conjunctive queries, as well Max-SAT and clause CONFIDENCE/SUPPORT functions. KnoBAB consists of a data loader and indexer, query compiler, and an execution engine, thus fully matching the architecture of a relational database. This solution was enabled by the extension of the traditional $\mathrm{LTL_f}$ operators, providing algebraic semantics to declarative temporal models, so as to support data operations over tuples representing trace events. Our solution is not limited to one single declarative language of choice, as it might support any possible model that can be expressed via $\mathtt{xt}\mathrm{LTL_f}$ operators. Based on the latest solutions in current database literature, the query plan was also designed to minimize the data access by running the common sub-queries at most once. KnoBAB outperforms state of the art solutions both tailored to the specific dataset or based on traditional relational databases running SQL queries. This solution will enable us to learn models exploiting abductive reasoning rather than traditional mining techniques, thus also providing safety guarantees over noisy data and models that are inconsistency free [18].

Future works will provide extensive benchmarks for bigger log datasets and will provide speed-up results for the parallelized execution of the resulting query plan: despite this being already implemented, we postpone those results due to the lack of space in the present paper. For the time being, the logs available from the research community are quite compact, and therefore the whole dataset is well fit in primary memory. Dealing with actual big data solutions or bigger models will require us to migrate the data store location to secondary memory, thus requiring the adoption of Near-Data Processing techniques [5]. As part of the data-loading phase, HUMAN READABLE LOG FORMAT key names currently only support strings consisting of letters. A proposed extension would allow for any possible string name, including numbers and symbols.

The adoption of relational databases and operator-based query plans might enable incremental trace updates so to extend those at runtime: this open research problem can be now solved by exploiting algebraic rewriting rules similar to the ones from relational databases, thus requiring a formal definition of $\mathtt{xt}\mathrm{LTL_f}$ operators.

## ACKNOWLEDGMENTS

Samuel Appleby's work is supported by Newcastle University.

## REFERENCES

[1] Giovanni Acampora, Autilia Vitiello, Bruno Di Stefano, Wil van der Aalst, Christian Gunther, and Eric Verbeek. 2017. IEEE 1849: The XES Standard: The Second IEEE Standard Sponsored by IEEE Computational Intelligence Society. *IEEE Comp. Int. Mag.* 12, 2 (2017).

[2] Simone Agostinelli, Giacomo Bergami, Alessio Fiorenza, Fabrizio Maria Maggi, Andrea Marrella, and Fabio Patrizi. 2021. Discovering Declarative Process Model Behavior from Event Logs via Model Learning. In *ICPM 2021*. IEEE, 48–55.

[3] Adriano Augusto, Ahmed Awad, and Marlon Dumas. 2021. Efficient Checking of Temporal Compliance Rules Over Business Process Event Logs. *CoRR* abs/2112.04623 (2021).

[4] Ladjel Bellatreche, Mohamed Kechar, and Safia Nait Bahloul. 2021. Bringing Common Subexpression Problem from the Dark to Light: Towards Large-Scale Workload Optimizations. In *IDEAS*. ACM.

[5] Giacomo Bergami. 2021. On Efficiently Equi-Joining Graphs. In *IDEAS*. ACM.

[6] Giacomo Bergami, Chiara Di Francescomarino, Chiara Ghidini, Fabrizio Maria Maggi, and Joonas Puura. 2021. Exploring Business Process Deviance with Sequential and Declarative Patterns. *CoRR* abs/2111.12454 (2021).

[7] Giacomo Bergami, Fabrizio Maria Maggi, Andrea Marrella, and Marco Montali. 2021. Aligning Data-Aware Declarative Process Models and Event Logs. In *Business Process Management*. 235–251.

[8] Andrea Burattin, Fabrizio Maria Maggi, and Alessandro Sperduti. 2016. Conformance checking based on multi-perspective declarative process models. *Expert Syst. Appl.* 65 (2016), 194–211.

[9] Eduardo González López de Murillas, Hajo A. Reijers, and Wil M. P. van der Aalst. 2022. Data-Aware Process Oriented Query Language. In *Process Querying Methods*. Springer, 49–83.

[10] Valeria Fionda, Gianluigi Greco, and Marco Antonio Mastratisi. 2021. Reasoning About Smart Contracts Encoded in LTL. In *AIxIA*. Springer International Publishing, Cham, 123–136.

[11] Stratos Idreos, Fabian Groffen, Niels Nes, Stefan Manegold, K. Sjoerd Mullender, and Martin L. Kersten. 2012. MonetDB: Two Decades of Research in Column-oriented Database Architectures. *IEEE Data Eng. Bull.* 35, 1 (2012), 40–45.

[12] Nathan John, Jeremy Gow, and Paul Cairns. 2019. Why is debugging video game AI hard?. In *Proc. AISB AI & Games symposium*. 20–24.

[13] Jingrui Li, Kento Goto, and Motomichi Toyama. 2021. SSstory: 3D data storytelling based on SuperSQL and Unity. In *IDEAS*. ACM, 173–183.

[14] Jianwen Li, Geguang Pu, Yueling Zhang, Moshe Y. Vardi, and Kristin Y. Rozier. 2020. SAT-based explicit LTLf satisfiability checking. *Artificial Intelligence* 289 (2020), 103369.

[15] Felix Mannhardt, Massimiliano de Leoni, Hajo A. Reijers, and Wil M. P. van der Aalst. 2016. Balanced multi-perspective checking of process conformance. *Computing* 98, 4 (2016).

[16] Youichiro Miyake. 2017. Current Status of Applying Artificial Intelligence in Digital Games. In *Handbook of Digital Games and Entertainment Technologies*. Springer Singapore, 253–292.

[17] André Petermann, Martin Junghanns, Robert Müller, and Erhard Rahm. 2014. FoodBroker - Generating Synthetic Datasets for Graph-Based Business Analytics. In *WBDB'14*.

[18] Jose Picado, John Davis, Arash Termehchy, and Ga Young Lee. 2020. Learning Over Dirty Data Without Cleaning. In *SIGMOD Conference*. ACM, 1301–1316.

[19] Paul Pichler, Barbara Weber, Stefan Zugal, Jakob Pinggera, Jan Mendling, and Hajo A. Reijers. 2011. Imperative versus Declarative Process Modeling Languages: An Empirical Investigation. In *BPM Workshops*. 383–394.

[20] Artem Polyvyanyy, Arthur H. M. ter Hofstede, Marcello La Rosa, Chun Ouyang, and Anastasiia Pika. 2019. Process Query Language: Design, Implementation, and Evaluation. Vol. abs/1909.09543.

[21] Marcella Rovani, Fabrizio Maria Maggi, Massimiliano de Leoni, and Wil M. P. van der Aalst. 2015. Declarative process mining in healthcare. *Expert Systems with Applications* 42, 23 (2015), 9236–9251.

[22] Anne Rozinat and Wil M. P. van der Aalst. 2008. Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* 33, 1 (2008).

[23] Stefan Schönig. 2015. SQL Queries for Declarative Process Mining on Event Logs of Relational Databases. *CoRR* abs/1512.00196 (2015).

[24] Stefan Schönig, Andreas Rogge-Solti, Cristina Cabanillas, Stefan Jablonski, and Jan Mendling. 2016. Efficient and Customisable Declarative Process Mining with SQL. In *CAiSE*. Springer.

# The adaptation of the OODA loop to the decision-making systems processing Big Data in the area of morality

Damian T. Węgrzyn
Polish-Japanese Academy of Information Technology
Warsaw, Poland
damian@wegrzyn.info

## ABSTRACT

The rapid development of autonomous systems and their presence in human life force them to make quick decisions based on Big Data. Many of these decisions involve moral judgments that are then transformed into specific actions. Side effects of the choices made by the decision systems can be dangerous, so we have to be very careful when increasing the capacity of these systems. The decisions the autonomous systems make should be as ethical as possible.

This paper adapts the observe-orient-decide-act (OODA) loop to the decision-making process in the moral area. It combines the parameterization of cognitive aspects of autonomous systems with ethical standards and moral inference. Problems related to the implementation of moral inference to autonomous systems, including artificial intelligence (AI) systems, are presented. Thanks to the adaptation of the OODA loop, it is possible to make morally correct decisions and actions based on a set of ethical principles adjusted to a specific situation.

The presented proposal allows for moral inference, which extends the possibilities of autonomous systems that use the inference loop, especially those processing Big Data. The decision-making system still has the possibility of a choice aimed at doing more good or less evil.

## CCS CONCEPTS

• **Applied computing** → **Multi-criterion optimization and decision-making**; • **Information systems** → *Decision support systems*; • **Computing methodologies** → *Knowledge representation and reasoning*; • **Computer systems organization** → Robotic autonomy.

## KEYWORDS

OODA loop, Big Data, Autonomous systems, Decision-making process

**ACM Reference Format:**
Damian T. Węgrzyn. 2022. The adaptation of the OODA loop to the decision-making systems processing Big Data in the area of morality. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3548785.3548806

## 1 INTRODUCTION

Nowadays, man and autonomous decision-making systems often follow the same paths. Intelligent machines can be found in many areas of human life: in medicine, business, and everyday life, e.g. autonomous vehicles. In many cases, these devices make the decisions. Some of them have moral consequences. In such cases, the decision-making system should conclude before taking action whether the decision and its implications are ethical.

Conducting a moral discourse in the area of information systems requires the analysis of several topics [17], including current ethical standards, intelligence, and morality, applied to machines, as well as the personality of AI systems, and ultimately understanding how self-learning decision systems process collected Big Data.

This paper contributes to the area of the decision-making process of autonomous systems using Big Data. It indicates problems related to the implementation of moral inference elements to AI systems. As one of the solutions, it proposes to adapt the OODA loop as a cycle that allows for simple moral reflection in a specific situation. The OODA loop is currently used in Big Data analysis to boost the outcomes [21], as well as to accelerate decision-making processes [7]. The presented considerations are also intended to encourage researchers to adapt and implement the elements of morality in the reasoning algorithms applied in the autonomous devices.

The paper is organized as follows. Section 2 presents the OODA loop and the main problems of its usage in autonomous systems. Section 3 describes the concept of general AI. Section 4 classifies the decision-making systems in the structure of intelligence. Section 5 presents problems with modern ethical standards. The relations of morality to the autonomous systems are presented in Section 6. Section 7 shows the issue of the electronic personality of AI systems. The proposal for the adaptation of the OODA loop to the decision-making process in the area of morality is presented in section 8. Section 9 is a discussion and comparison of the results obtained in other works. The outcomes of the research are presented in the Conclusions section.

## 2 OODA LOOP IN THE DECISION-MAKING PROCESS OF AUTONOMOUS SYSTEMS

The OODA loop is an acronym for four activities: observe, orient, decide and act. This cycle connects acquisition and analysis of information, decision selection, and action. Although this solution was originally designed for military purposes, it is currently used, i.a. in describing learning processes [2]. This methodology is suitable

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548806

for describing the decision-making process performed by a single actor.

## 2.1 OODA loop stages

The first stage of the OODA loop is observation, i.e. obtaining information from the environment. It is usually carried out by various sensors that replace human senses. At this stage, an important issue is the vulnerability of the systems to unreliable or intentionally crafted information. In the case of autonomous systems, it cannot be limited to receiving information only from selected sources, as this may result in the exclusion of relevant information. This stage is crucial because conclusions and decisions are made on these data. Alternatively, the information held by the system can be supplemented with new data collected from the environment. Another important part of this stage is to define and determine the parameters that will be used in the next step of the loop [3].

The second stage is the orientation, i.e. synthesis and analysis of the collected information, in order to build an up-to-date perspective of the situation. Autonomous systems do this with the use of weights that are assigned to individual parameters. The weights are usually set by the manufacturers. The user can also define weights for individual parameters if the producer allows it. At this stage, crowdsourcing is also used to distinguish between weight values [26].

The third step is to choose a decision, which consists of determining the actions that should be performed based on the orientation in a situation. Usually, there are several possible choices, and the probability of success is determined for each of the possibilities. The decision algorithm is based on the potential success and less damage [3]. Problems arise when making decisions related to fuzzy attributes, such as giving joy or pain. In this case the producers shift the final decision to the human being.

The last stage of the OODA loop is an action. It consists of implementing the selected choice of action. This usually boils down to direct interaction with the environment.

## 2.2 The problems of using the OODA loop in autonomous systems

The use of the OODA loop in autonomous systems causes problems. One of them is the reduced control of the device and the predictability of its decisions in the case of multiplication of the loop usage [15].

The first problematic area is the initial stage of OODA, i.e. obtaining information. First of all, receiving information from other systems or external sensors is problematic In such a situation, it is difficult to make sure that the obtained data is correct; it can only be trusted. In the case of already collected data, it is necessary to keep them up-to-date. Not only the data itself become outdated, but also the context and relations between the data become outdated, too.

In the second stage, the problem is the adequate assignment of weights to various data collected for inference. It is possible to create general classes of collected data and the weight ranges that can be assigned within the classes, but the weight determination has to be done dynamically, ad hoc. Predefining weights do not work well in dynamic situations and the context, which may also cause differences in weights assigned to individual parameters.

In the third stage, i.e. in the decision-making step, attention should be paid to the problem of inference from a large number of parameters. Some parameters are undefined and appear as new in the previous stage of the loop. Various interpretations of the existing situations and unclear norms or practices concerning the same area of interest also pose a problem. Ambiguity in the case of algorithms also causes complex problems in autonomous decision-making systems. Another problem at the decision stage is the need for efficient decision-making. It takes time to collect data, interpret it, assign weights, and analyze it. There are situations, when a decision should be made extremely quickly, and it will be of significant consequences. This is a performance challenge for AI machines on the one hand, and an optimization challenge for their manufacturers on the other.

# 3 THE IDEA OF GENERAL ARTIFICIAL INTELLIGENCE

The concept of intelligence is nowadays defined as the ability to perceive, analyze and adapt to changes in the environment, or as the ability to understand, learn and use the knowledge and skills in various situations [20]. Psychologists of the $20^{th}$ century distinguished many types of intelligence [6], such as social or motor intelligence. The definition of AI appeared in the 1950s [16], when the subject has been transferred from man to machine. From that moment on, AI and natural intelligence should be distinguished. The main goal of AI engineers is to create systems with human-like intelligence, although it is not the only type of intelligence.

AI, similar to human intelligence, is being developed today in two areas [26]. The first of them is narrow AI, which maps the features of human intelligence only in predefined ranges: actions, choices, or situations. However, in addition to devices that are adapted to actions and behavior only under certain conditions, there is a need for devices that are close to the second category, the so-called general AI. So far, it has not been possible to produce general AI by artificial means, mainly from technical constraints. Nevertheless, some features of natural intelligence are possible to design and implement [11].

Both human beings and devices share common elements. The key common point is collecting information about the context (environment, conditions, situation, etc.), processing it, and making decisions and actions based on the information processed. When designing devices with AI, engineers have to consider the assumption that an abstract approach to intelligence is never separated from emotions, the environment, and other people [26]. Therefore, this issue has to be approached holistically, just as a human being perceives the reality and functions in it.

# 4 THE SELF-LEARNING AUTONOMOUS SYSTEMS

The way to create general AI is most evident in the generation of devices that pretend to be fully autonomous systems. When analyzing this group of devices, we should start with the broadest type, namely interactive systems. These systems actively and spontaneously interact with their environment. They often enter the sociological sphere and everyday life of a person. Going further, such devices as flying or driving vehicles initiate their operation by themselves, and are often able to modify the rules of their actions based on information received from the environment. Self-learning autonomous systems are additionally able to collect and process information about the environment and the context of the situation, and to modify the models of their activities. Decision-making systems, which additionally make important decisions based on newly acquired information, are a special case of such systems. The hierarchy of systems in the area of intelligence is shown in Figure 1.

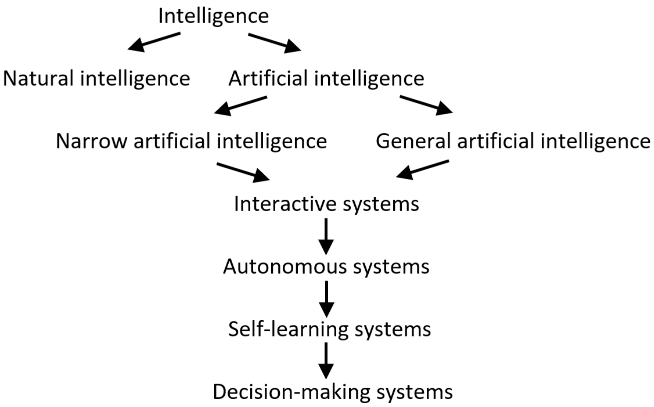


**Figure 1: The autonomous decision-making systems in the area of intelligence.**

Although there are no completely autonomous decision-making systems, there are modern devices that are confronted with situations requiring a choice based on a new context. The dynamic technological development has the potential to create independent systems [14]. This raises many problems related to the responsibility for the choices and actions taken, as well as the moral evaluation of actions undertaken. To simplify the procedures, it might be recommended to treat AI systems as a product [14]. In addition, the primary responsibility for the operation of AI devices is put on software developers [24]. In the case of autonomous decision-making systems, it is not known what conclusions - and therefore actions - the machine can ultimately reach. Currently developed decision-making systems are usually based on machine learning, and especially on deep learning technology, which can often lead to unexpected behavior [14]. The more freedom in making decisions an AI device receives, i.e. the greater the level of machine autonomy, the more hazards it may generate. Such software is highly unpredictable, as it is based largely on ad-hoc data, and as a result the probability of obtaining completely unexpected or even unintended outcomes is increased. In many difficult situations, it is recommended to transfer the decision to a person who has the capability to take control of the AI system operation at any time, especially when the operation of the machine may cause damage or threat to human health or life [13]. Some authors are inclined to the thesis that man should be the last link of every decision-making chain [12, 22]. To evaluate certain behaviors, there is a need for a moral judgment to decide whether man should take control of the device [29]. The doubts arising from the unclear relationship between the device and the damage triggered movements undertaken by legal regulators to eliminate the problem of "distributed irresponsibility" [14].

## 5 THE PROBLEMS OF CONTEMPORARY ETHICAL STANDARDS

Contemporary regulations are trying to keep up with the dynamically developing technological progress. In the past decade, a lot of work has been undertaken to eliminate the dissonance between the law and the actual usage of autonomous systems. However, the law still fails to guarantee fair regulations in the futuristic context of AI [9, 14].

Many problems arise in the process of creating new, universally applicable standards in the field of autonomous decision-making systems. The first problem is delegating decisions. The questions concern determining the scope of the decision, responsibility for the decisions made, as well as the comprehensibility of the reasons for the decision itself. Another problem is injustice and social inequality. In the case of self-learning systems, a frequent case is the bias of algorithms that ends up with biased reasoning, which affects the decisions made [18]. Personal data protection also causes many problems regarding the tracking of user preferences, and their disclosure directly or indirectly [4]. Users are often treated subjectively, which violates human dignity. Another example of problems is the methodology of the implementation of ethical norms in algorithms: what source of ethical norms should be adopted, which norms are obligatory for particular geolocations, and who is ultimately responsible for the operation of the device in the situation of a breach of the adopted ethical norms.

In commonly known ethical standards [1, 5, 10, 12, 25] it is possible to distinguish some common groups of values, mentioned in most norms. Some of them are a big challenge for AI device manufacturers; the transparency of choices and actions is a good example of these challenges. This problem is known as the "Black box" [28]. The premises of the reasoning in neural networks algorithms that AI systems usually rely on are not clear. Therefore, ethical standards also raise the issue of the clarification of the decision-making process. From the user's perspective, it should be intelligible, and from the producer's and developer's perspective, it should be accountable. The next value is justice of the choices made, but, as mentioned above, there is no rigid framework for this, although it is already being implemented in some areas [8]. The standards implement it differently, namely, non-maleficence and doing beneficence are recommended to be implemented by developers. The standards also remind producers about the privacy of data processed using AI systems and about data anonymization, if possible. An important problem is to determine who is responsible for the choices and actions of AI systems, especially when it comes to autonomous and adaptive technologies. The problem is how to distribute responsibility between the creators, operators, and users of AI systems.

The creation of ethical standards and regulations usually encounters various problems in terms of their application by manufacturers. The first problem is the selectivity in choosing the rules to implement (ethics shopping), i.e. some rules are implemented, while the others are ignored. Another problem is ethics bluewashing, i.e. creating standards or rules that have no real impact on the behavior of the device, just to show the use of ethical standards in created AI systems, in order to improve the brand or product image. Another example is ethics lobbying, when the created rules are beneficial for a given producer, region, or nation. A similar problem is ethics dumping. Since imposing ethical standards is seen as a restriction to devices, the competitive manufacturers perform ethics dumping by asserting that they do not restrict their devices ethically (by applying ethical standards) as other manufacturers. The last problem is ethics shirking, i.e. not applying the ethical norms and principles, because no severe penalties are applied for such a practice, or obeying ethical standards is not enforced.

## 6 MORALITY IN THE CONTEXT OF AUTONOMOUS SYSTEMS

The philosophy of AI considers whether intellect and consciousness can be assigned to an autonomous machine, if the machine works in a way similar to human behavior [19]. This assumption is a step towards granting morality to such AI systems, but this poses a wider problem than their mere awareness. Still, this does not diminish the moral nature of decisions and actions taken by autonomous, decision-making AI systems. Since the machine with AI can make decisions that are subject to moral evaluation, it becomes an actor subject to moral evaluation as well.

The seemingly trivial moral judgments may often have hidden complexity. Although AI systems are technologically advanced, there are beliefs [22] that they will not be able to make morally correct decisions as they will not be capable of ethical reflection. In a situation of moral dilemmas, when there is no unambiguous, ethically justified solution, and various ethical theories propose different solutions, reasonable choice should be made, justified, and explained. This imposes flexibility in AI decision-making systems, as sticking to only one ethical theory may lead to the acceptance of an undesirable pattern of moral behavior, i.e. to puritanism [22]. Thus, it is impossible to adopt one ethical theory and order its implementation by ethical standards in AI systems, because the very choice of such a theory is already a moral choice, determining the concepts of good and evil. Moreover, there will be no choice here.

According to the current state of art, AI is neither emotional nor moral intelligence. Choosing the proper solution is related to the common sense, which is realized in making the right choices, and which only characterizes people [22]. Reasoning is also not the feature of AI systems. The decisions of AI machines are, at best, rational. In making the proper decisions in the moral sense, three values should be followed: righteousness (compliance), forbearance, and prudence [22]. All these features are attributed only to man: forbearance, awareness of the limits of the intelligence, and the prudence of discerning what decisions are the most appropriate. Having moral intelligence is related to the ability to make judgments based on the intellectual and moral virtues that constitute the personality of the subject. Additionally, moral intelligence represents the ability to be honest, responsible, empathetic, and forgiving [22]. General theology connects morality, self-awareness, interaction, and even love in a man who is the image of God [26].

A man has the ability to limit the wide area of analysis, using various relations, contexts, or shortcuts, thus setting the framework to the decision process [26]. A similar selection should be made by the decision-making system, as otherwise the system will not be able to process a huge amount of data, or the decision-making process will be stretched over time. Man uses his mind in a highly selective way, whereas the machine processes all the information it obtains. The selection skills of man are shaped through interactions with the environment, and the acquisition of such skills is a long-term process and requires a social context. This entails such skills as recognizing emotions, compassion, and understanding. Establishing and maintaining interpersonal interactions, identifying social roles or practices, and managing one's behavior are the consequences of moral intelligence, too.

The theological approach to AI systems, assuming they are similar to a man and thus as the image of God, imposes selected abilities on the subject, such as the ability to think abstractly, behave morally, maintain a relationship with the environment (following the formula of divine relations), transcend oneself, or having a sense of freedom. The emotional area of AI systems may show inauthenticity, simulation, or even a lack of empathy. This is problematic because the behavior of AI systems may shape the general moral status in the future, as they will become part of a system that sets certain limits of behavior.

It must be admitted that any information technology is not morally neutral. When making decisions, it implements acts that are not amoral. Thus, there is a chance to implement at least a moral base in autonomous decision-making systems, because the basic moral principles do not depend essentially on external factors, such as the decisions of society: evil is evil even if everyone does evil, and good is always good, even if no one does well.

## 7 TOWARDS THE ELECTRONIC PERSONALITY OF AI SYSTEMS

The development of AI is approaching a limit that is difficult to exceed, namely the self-awareness of AI systems. Currently, the sense of one's existence is attributed only to man. Nevertheless, research on AI does not exclude the emergence of machines with consciousness, capable of non-obvious thinking.

Nowadays, AI successfully models abstract and conscious human reasoning, relying on the accepted principles of logic. These are only some selected functions of the human mind that man performs in an automated manner, without further reflection on them, in a sensorimotor way [26]. The producers of autonomous decision-making systems face challenges such as empathizing systems, respecting dignity, and the desire of good for other individuals. Other challenges are forbearance, understanding, discovering hidden meaning, and significant social interactions.

Unlike consciousness or self, personality can be determined using technical parameters such as temperament levels. The topic of electronic personality (e-person) is raised in the discussions on AI systems. Furthermore, emphasis is placed on the deontological dimension of a subject that respects the adopted ethical standards in decision-making processes. Contemporary regulations place AI systems in the area of digital ethics, for which dignity in the deontological understanding is the center [14].

# 8 ADAPTATION OF THE OODA LOOP TO THE DECISION-MAKING PROCESS IN THE AREA OF MORALITY

In the case of making moral autonomous decisions, we deal with the concept of moral intelligence. This concept is based on the feeling of moral consequences related to the decisions made. It raises many ethical problems and confusion when it comes to moral dilemmas. Moral intelligence also implies having a basic knowledge of ethics, which allows making choices following ethical principles.

The first step in adapting the OODA loop to ethical decisions is the implementation of ethical principles, practices and standards as a permanent component. This adaptation takes place in the second stage of the loop, when setting parameters and weights for individual information collected in the first stage of the loop. It would be a huge risk to let self-learning algorithms independently define basic moral concepts, because it may lead to cognitive distortions. The limitations of algorithms, pose problems in defining such concepts as good or luck. When trying to define these concepts as primary, we come to the problem of regressus ad infinitum, namely the subsequent concepts have to be defined, and so on. The fuzzy set theory can be a solution to this problem, as this theory can deal with imprecise information, and provide precise output needed for the decision algorithms [27]. Creating a predefined moral framework allows for autonomous and free decisions, albeit with one limitation: decisions and actions are aimed at producing the greatest possible good or the least evil. Therefore, it is a model of moral acceptability [22], used in ethical theories. Moreover, this paper assumes that the data that is being retrieved in this stage is reliable. The issue of quality of the collected data from various sensors is a complex issue, analyzed in contemporary research [23] although it is not the subject of this research. Undoubtedly, this is an issue for future consideration to improve moral inference.

To adjust the inference stage, the essence of the moral decision and the ethical context, that influences it, have to be understood. Moral consideration comes down to a multifaceted approach to the situation and the selection of a set of specific moral principles that are the reference. As mentioned, they should be predefined in the decision-making system, although many ethical approaches may be implemented. The selection of the most suitable among them may be made by the AI system in the second stage of the OODA loop, recognizing the situational context. However, in this case, it is not possible to rely on the principles developed from previous loop outcomes. In line with the unpredictability in the decision-making process, they can lead to immoral decisions. It should be noted that predefining ethical principles, on the one hand, allows maintaining high morality of conclusions, and on the other hand, prevents subjective verification or creation of machine's criteria for moral decisions.

The second stage of the OODA loop should be extended with weight classes, resulting from the significance of data that can be assigned with different values, but only in the areas of predefined ranges. These ranges will be adapted by the decision-making system to a specific situation or context and will result from the adopted ethical norms and principles. Predefining the weight classes depends on the manufacturer, responsible for the level of freedom that will be left to the system in making decisions. Therefore, the autonomy of the decision-making system lies only in the range allowed by the manufacturer. Such a reduction should be seen as the virtue of the system. There are also such good limitations in nature, e.g. limiting the killing of other creatures (with the general assumption of species survival).

To sum up, the approach to the OODA loop, proposed in this paper, allows the usage of this methodology in the area of moral inference. This requires extending the loop in three stages. In the first stage (observation), various ethical standards and norms should be implemented. In the second stage (orientation), the parameters and weight ranges, corresponding to them, should be defined, e.g. using fuzzy sets. Finally, the third, decision-making stage, should contain the choice of a specific ethical theory, based on the situational context.

The proposed adaptation can be used in many autonomous systems that collect a lot of data from the environment and must efficiently make decisions. Self-propelled vehicles or autonomous robots are good application examples. The proposed solution can also be used in automatic systems, such as evaluation systems (i.a. of prisoners), moderating social media posts, or systems for recognizing emotions and moral attitudes.

# 9 DISCUSSION

K.A. Chagal-Feferkorn [3] described the use of OODA loops in decision algorithms. The author distinguishes between criteria that must be met by decision-making systems of various categories, e.g. life and death decisions, decisions requiring real-time inference, or the dynamic nature of the sources, from which the system obtains information. In addition, to assess the decision-making autonomy of the AI system, human-based metrics are used in order to assess to what extent the device can replace man and what the device's freedom of operation looks like. Such metrics depend on many elements at individual stages and therefore should be considered in a specific stage of the decision loop.

The use of any decision-making framework by AI systems brings problems. E. Magrani [14] rightly indicates the values that should be taken into account in the preparation of such systems: fairness, reliability, security, privacy, data protection, inclusiveness, transparency, and accountability. Nowadays, it is not always possible to provide all these attributes at the same time, although this cannot be an excuse for abandoning any of them. The provision of the indicated values ultimately depends not on the loop itself, but on the methods used at its stages in the process of implementing the functionality. Therefore, mentioned attributes are important guidelines, but they do not affect the methodology of the OODA loop. As rightly noted by E. Magrani, the designed models should be focused on a human being. They should be sensitive to values, which is ensured by the OODA loop in the case of taking into account the adaptation requirements, described in Section 8.

From the ethics point of view, using the OODA loop causes dealing with heuristics. G. Szulczewski [22] therefore sees a problem related to the unpredictability of moral reflection in the decision-making process. Additionally, he notices that in many situations there is no time to perform analyzes. These problems are real, but they are caused by the decision-making process itself not by moral analysis. The advantage of decision-making systems over a man should be noticed here, as the systems can use a lot of information that would require decades for a person to learn, or even would never be learned.

Undoubtedly, the use of the OODA loop in decision-making processes, including moral choices, will not provide autonomous systems with features of the moral intellect of a human being, such as moral scruples, psychological discomfort, or compassion. These attributes influence a person's moral decisions, as they are used as guidelines to make the correct choice. Currently, no system can implement such features, because they are associated with the self-awareness of existence. Still, morally correct choices can be made in a specific situation, which is denied by G. Szulczewski [22]. Indeed, this is still an adaptive decision-making process, albeit leading to a morally correct decision and therefore moral action.

The use of models that allow moral inference, as rightly noted by E. Magrani [14], obliges us to consider such systems as the so-called moral machines that actively participate in society. This entails important challenges for producers of autonomous decision-making systems, especially their responsibility for the machines that constitute human reality. The use of various moral models, including the OODA loop, allows avoiding many immoral behaviors, which is always an added value for the environment participating in the interaction.

## 10 CONCLUSIONS

The OODA loop is a proven solution used in autonomous systems processing Big Data. The application of the adaptive assumptions, proposed in Section 8 of this paper, allows the use of this cycle also for moral inferences. In some circumstances, the OODA cycle works better than humans, as it can use a lot of information that a person is unable to learn in his life.

The use of any solution does not change the fact that we are still dealing with a limited machine, although adapted to autonomous functioning in society. Contemporary autonomous decision-making systems only pretend to possess unattainable human attributes, such as consciousness, but in most cases, they do not need it. It is enough for them to be guided by superior, desirable ethical values in their interactions with the environment.

While it is currently impossible to build moral intelligence, there are no barriers to scientific development. The use of various solutions in the decision-making processes of ethical (at least to some extent) AI systems is a step towards creating the volitional area of machines. The OODA loop is a very good example that, on the one hand, it is possible at all, and on the other hand, it does not require a lot of work. The way to achieve the goal is to adapt to human decision-making in the areas of morality.

## 11 ACKNOWLEDGMENTS

The author would like to thank Professor Alicja Wieczorkowska for invaluable help in commenting and corrections to this paper.

## REFERENCES

[1] *British Standards Institute.* 2016. *BS 8611:2016. Ethical design and application of robots.*

[2] Ian T. Brown. 2018. *A new conception of war: John Boyd, the US marines, and maneuver warfare.* Marine Corps University Press, Marine Corps Base Quantico, VA, USA.

[3] Karni A. Chagal-Feferkorn. 2019. *Am I an algorithm or a product: when products liability should apply to algorithmic decision-makers.* Stanford Law & Policy Review 30, 61-114.

[4] Nicola Fabiano. 2019. *Ethics and the protection of personal data.* Journal on Systemics, Cybernetics and Informatics 17, 2 (2019), 58-64.

[5] *Federal Ministry of Transport and Digital Infrastructure of the Federal Republic of Germany.* 2017. *Report by the Ethics Commission on Automated and Connected Driving.* June.

[6] Horward E. Gardner. 2000. *Intelligence reframed: multiple intelligences for the $21^{st}$ century.* Hachette, UK.

[7] Bryan Harris. 2014. *Closing the OODA Loop: Using Big Data and Analytics to Improve Decision Making.* SAS Global Conclusions Paper, 2 (2014).

[8] Gry Hasselbalch. 2021. *A framework for a data interest analysis of artificial intelligence.* First Monday 26, 7 (2021), https://doi.org/10.5210/fm.v26i7.11091

[9] Patrick Henz. 2021. *Ethical and legal responsibility for artificial intelligence.* Discover Artificial Intelligence 1, 2 (2021), https://doi.org/10.1007/s44163-021-00002-4

[10] *IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems.* 2019. *Ethically aligned design.*

[11] Müge Karabağ. 2021. *A theoretical overview of artificial intelligence ethics within the context of coding moral values.* TRT Akademi 6, 13 (2021), 748-767, https://doi.org/10.37679/trta.954641

[12] *Korea's Ministry of Commerce, Industry and Energy.* 2012. *South Korean robot ethics charter.*

[13] Utku Kose, Ibrahim Adra Cankaya, and Tuncay Yigit. 2018. *Ethics and safety in the future of artificial intelligence: remarkable issues.* International journal of engineering science and application 2, 2 (2018), 65-70.

[14] Eduardo Magrani. 2019. *New perspectives on ethics and the laws of artificial intelligence.* Internet Policy Review 8, 3 (2019), 1-19, http://dx.doi.org/10.14763/2019.3.1420

[15] Eduardo Magrani, Priscilla Silva, and Rafael Viola. 2019. *New perspectives on ethics and responsibility of artificial intelligence.* In (Eds.) Caitlin Mulholland, and Ana Frazao. *Artificial intelligence and law: ethics, regulation and responsibility.* Thomson Reuters, Revista dos Tribunais, São Paulo, 117.

[16] John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude E. Shannon. 1959. *Artificial intelligence.* Research Laboratory of Electronics Progress Report 53.

[17] Zorica Mijartovic, and Orhan Jašić. 2021. *Ethics for "intelligent" beings created by man: scenarios of the future.* Epistēmēs Metron Logos 6, 6 (2021), 22–29, https://doi.org/10.12681/eml.27743

[18] Ellen Pearlman. 2021. *Building a 'Sicko' AI: AIBO: An emotionally intelligent artificial intelligent GPT-2 AI Brainwave Opera.* In ACM International Conference on Interactive Media Experiences (IMX '21), New York, NY, USA, 205–207, https://doi.org/10.1145/3452918.3467814

[19] (Eds.) Stuart Russel, and Peter Norvig. 2003. *Artificial intelligence: a modern approach.* Prentice Hall, Haboken, NJ, USA.

[20] Jan Strelau. 1997. *Human intelligence.* ŻAK, Warsaw, Poland.

[21] Robert Szeligowski. 2018. *Cognifying the OODA Loop: Improved Maritime Decision Making.* Gravely Naval Research Group, Naval War College.

[22] Grzegorz Szulczewski. 2019. *Artificial intelligence and moral intelligence. An introduction to cybernetic ethics.* Annales. Etyka W Życiu Gospodarczym 22, 3 (2019), 19–31, https://doi.org/10.18778/1899-2226.22.3.02

[23] Hui Yie Teh, Andreas W. Kempa-Liehr, and Kevin I-Kai Wang. 2020. *Sensor data quality: a systematic review.* Journal of Big Data 7, 11 (2020), https://doi.org/10.1186/s40537-020-0285-1

[24] *United Nations Commission on International Trade Law.* 2005. *United Nations Convention on the Use of Electronic Communications in International Contracts.* 23 November. Vienna, Austria.

[25] Gianmarco Veruggio. 2006. *The EURON Roboethics Roadmap.* In *$6^{th}$ IEEE-RAS International Conference on Humanoid Robots*, 612-617, https://doi.org/10.1109/ICHR.2006.321337

[26] Aku Olavi Visala. 2018. *On the theology of artificial intelligence.* Teologinen Aikakauskirja 123, 5 (2018), 402-417.

[27] Damian Węgrzyn. 2021. *Towards improving the decision-making process of artificial intelligence devices in situations of moral dilemmas.* In (Eds.) Jorge P. Borondo, Mario A. Oliva, Kiyoshi Murata, and Ana M. L. Palma. *Moving technology ethics at the forefront of society, organisations and governments.* Universidad de La Rioja, Spain, 168-179.

[28] Alan Winfield. 2017. *ELS issues in robotics and steps to consider them. Part 3: ethics.*

[29] *World Commission on the Ethics of Scientific Knowledge and Technology.* 2017. *Report of COMEST on robotics ethics.* 14 September. Paris, France.

# Data Science Applied to Discover Ancient Minoan-Indus Valley Trade Routes Implied by Common Weight Measures

Peter Z. Revesz
School of Computing, University of Nebraska-Lincoln
revesz@cse.unl.edu

## ABSTRACT

This paper applies data mining of weight measures to discover possible long-distance trade routes among Bronze Age civilizations from the Mediterranean area to India. As a result, a new northern route via the Black Sea is discovered between the Minoan and the Indus Valley civilizations. This discovery enhances the growing set of evidence for a strong and vibrant connection among Bronze Age civilizations.

## CCS CONCEPTS

• **Information systems** → Information systems applications; Data mining.

## KEYWORDS

Data mining, Indus Valley civilization, Minoan civilization, Trade route, Weight measure

**ACM Reference Format:**
Peter Z. Revesz. 2022. Data Science Applied to Discover Ancient Minoan-Indus Valley Trade Routes Implied by Common Weight Measures. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3548785.3548804

## 1 INTRODUCTION

Discovering long-distance trade relations gives a deeper insight into the economies of ancient civilizations. For example, lead ingots were traded between Sardinia and Israel (Yahalom-Mack et al. [13]), and the Minoans on the island of Crete traded vervet monkeys and baboons with eastern Africa (Urbani, and Dionisios [12]) and cumin *(Cuminum cyminum)* with India (Tsafou and García-Granero [11]). Together with the exotic goods, their names also spread as loanwords [1]. However, exotic goods constituted only a small part of the trade among ancient civilizations. A more sophisticated view of the intensity of trade relations can be obtained by an analysis of the weights that were used at various locations.

Recently, Ialongo et al. [4] published an analysis of the Bronze Age weight system and argued that an essentially common weight system spread from Mesopotamia to the west all the way to Ireland and to the east all the way to the Indus Valley Civilization. They gave a mathematical analysis that suggests that as merchants traveled from one place to another, they took their balance scales and weights with them and allowed the local merchants to copy these weights. Therefore, the main mode of weight exchange was successive copies being made throughout a huge trade zone that did not have a central authority over it. That is surprising and contradicts earlier assumptions that the introduction of a unified weight system requires a central authority that is intent to standardize trade within the realm of some kingdom or empire by fixing a standard weight to which every other weight must be adjusted.

Ialongo et al. [4] showed that while a uniform weight system could emerge without the intervention of a central authority, the successive copying of weights meant that the average unit weight gradually shifted from the original Mesopotamian unit as the use of the weights spread to the peripheries.

Instead of taking an overall view of the spread of the Bronze Age weight system, in this paper we focus on the Minoan weight system and try to answer the question of from where the Minoans acquired their weight measures.

The rest of this paper is organized as follows. Section 2 describes the data sources with a full listing of all the known weights that were used by the Minoans and others in the Near East and Middle East in ancient times. Section 3 presents the data mining results with the main discoveries of associations between the weights of several locations. Section 4 discusses the results in terms of geographical distribution and possible alternate trade routes that may have existed in the past. Finally, Section 5 gives some conclusions and describes future work.

## 2 DATA SOURCES OF WEIGHT MEASURES

Ialongo et al. [4] provided the exact measurement of 2274 Bronze Age stone, metal weights. They collected this large data set over ten years by visiting various museums and taking measurements of the weights contained in the collections of those museums. Table 1 shows that there are 71 Minoan weight measures from Crete and 112 Minoan weight measures from the Cyclades, which are represented by Akrotiri, Ayia Irini, and Philakopi. Hence there are a total of 183 Minoan weight measures within the large data set. Out of these four pairs have identical weights, which are highlighted in light green. We shifted some of the rows left or right to align the identical weights. Without double counting the identical weights, there are a total of 179 different Minoan weight measurements. When the exact locations of the Cretan weight measures are unknown, then the site name is simply indicated as 'Crete.' Similar data is available for the other Bronze Age sites in Ialongo et al. [4].

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548804

**Table 1: Weight measures in grams at various Minoan Bronze Age sites**

| Site | Weights |
|---|---|
| Akrotiri | 11.8, 12.7, 14.5, 16.3, 20.2, 23.2, 28.9, 32, 33, 35.25, 36, 37.8, 39, 39.65, 41.85, 42.5, 48.9, 52.9, 54.5, 56.6, 58, 65, 66.5, 80.3, 84, 86.1, 88.1, 88.3, 92.2, 95.4, 110, 115.3, 119.6, 169.7, 184.9, 187, 216, 234.5, 236.9, 239.9, 241.9, 252.5, 297.7, 357.8, 369.2, 483.8, 704.6, 744.3, 1009.4, 1021.2, 1028, 1162, 1408.4, 1506.2, 1619 |
| Ayia Irini | 12, 13.2, 13.6, 15, 15.2, 15.4, 15.9, 20.25, 20.35, 28.1, 28.6, 30.35, 31.1, 31.5, 31.6, 34.4, 34.9, 35.8, 38.7, 39.4, 39.7, 40.1, 42.3, 53.3, 54.8, 55.75, 57.9, 58.1, 58.3, 58.9, 59.95, 60.3, 61.15, 61.25, 63.6, 64, 64.9, 65.55, 67.05, 70.45, 79.9, 83.05, 85.7, 88.7, 91.9, 97.7, 121, 219.2, 390.6, 506.6, 626.7, 965.2, 1030.1, 1158.8 |
| Crete | 3.6, 7.5, 8.4, 43.25, 66.5, 73.62, 94, 113, 1140 |
| Haghia Triada | 24.3, 50.7, 237.1, 319, 402.9, 1487.8 |
| Katsambas | 9.3, 9.8, 10.2, 10.5, 11.4 |
| Knossos | 5.15, 8.45, 8.54, 12.6, 15.57, 16, 19.4, 19.82, 22.05, 35, 42.7, 59.92, 62.26, 96.4, 273.47, 327.02, 1567.47 |
| Mavro Spelio | 11.4, 57.4, 74.4, 251.8 |
| Mochlos | 19.4, 29.3, 30.4, 32.1, 43.7, 44.5, 92.9, 342.2, 720.3, 828.5, 14581.1 |
| Pachyammos | 31.7 |
| Palaikastro | 7.8, 14.4, 33.38, 63.1 |
| Philakopi | 190, 470, 1530 |
| Praisos | 46, 506.9 |
| Tylissos | 6.4, 9.5, 23.9, 30.6, 33.7, 40.8, 47.2, 220, 310, 472.4, 473.4, 477.5 |
| Zakros | 220, 1421.3 |

## 3 DATA ANALYSIS

According to Ialongo et al. [4], the weight measures were repeatedly copied as merchants spread the weight system wherever they traveled. Each copying could introduce some error. Hence if there is an exact match between two weights, then it is a strong indication that one is a direct copy of the other instead of being just a copy of a copy of some degree. In other words, perfect matches are indicators of direct trade links where the merchants took goods and their weights between the two locations. To investigate direct trade links, we selected three broad regions for our study:

1. The Minoan civilization, which flourished in the Bronze Age on Crete and the Cyclades.
2. The Fertile Crescent, which in our study included Mesopotamia, Syria, and southern Anatolia.
3. The Indus Valley civilization, which includes three major towns: Chanhu-Daro, Harappa, and Mohenjo-Daro.

We identified all perfect matches among the weights in the database that linked across at least two of these three regions and involved the Minoan civilization. Figure 1 shows these inter-regional matches. We found 30 inter-regional matching weight measure pairs, triangles and quadrangles that involved the Minoan civilization. Out of these 30 inter-regional matching weight groups, 26 groups, or about 86.7 %, agree with Ialongo et al.'s theory of a gradual spread from the Fertile Crescent area to both west to the Aegean area and west to the Indus Valley civilization. However, four groups, or about 13.3 %, do not seem to fit well into that theory. These groups, which area highlighted in pink in Table 2, are puzzling for the theory because they show matching weights between the Indus Valley and the Minoan civilizations without the common value occurring anywhere in the Fertile Crescent.

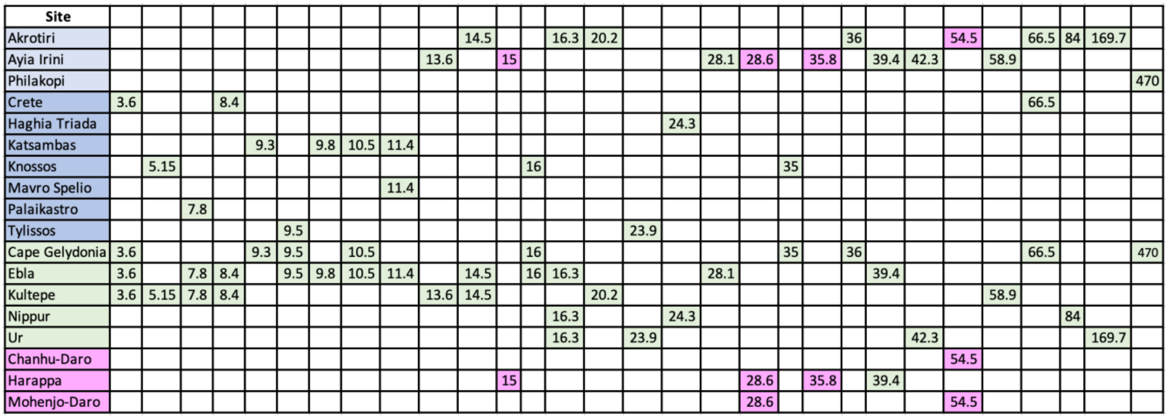
Consider for instance the 28.6 grams weight triplet. According to Ialongo et al.'s theory, a 28.6 grams weight had to exist somewhere in the Fertile Crescent, and from there merchants traveled to at least three different destinations (1) Ayia Irini, (2) Harappa, and (3) Mohenjo-Daro, where perfect copies of these were made.

If we were to accept Ialongo et al.'s theory, then we must assume that all the 28.6 grams weights that were used in the Fertile Crescent are now lost. That is unlikely. We also would have to accept that at three different times perfect copies were made of the 28.6 grams weight that is now lost. That is also unlikely.

The alternative explanation for the existence of the 28.6 grams weight triplet is that the original 28.6 grams weight existed in the Indus Valley civilization. Suppose that a Harappan merchant took it to Mohenjo-Daro, where a copy was made, and to Ayia Irini, where another copy was made. This explanation is more plausible than the previous one because it presupposes making two perfect copies instead of three perfect copies and presupposes that the weights survive in the place of origin instead of being lost.

Of course, even more explanations are possible. For instance, one may suppose that the original Fertile Crescent weight was the 28.4 grams weight found at Ur, and from there a merchant traveled to Mohenjo-Daro where a copy was made with some error that resulted in a 28.6 grams weight. Then a perfect copy was made at Harappa. Then the Mesopotamian merchant went to another trip to Ayia Irini, where again the same error of 0.2 grams was made, which again resulted in a 28.6 grams weight. The advantage of this explanation is that we no longer must account for a lost weight. On the other hand, it has only a small probability that the same magnitude copying error will be made at two different locations at two different times.

Any statistical model of the various scenarios to explain the data will have to rely on various assumptions about the probability of a certain weight measure being lost in a region and about the probability of copying errors of various sizes. Unfortunately, these assumptions may always remain questionable to those people who cannot imagine a direct trade route between the Indus Valley and



| Site | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Akrotiri | | | | | | | | | | | 14.5 | | | 16.3 | 20.2 | | | | | | | 36 | | | 54.5 | | 66.5 | 84 | 169.7 | |
| Ayia Irini | | | | | | | | | | 13.6 | | 15 | | | | | | 28.1 | 28.6 | | 35.8 | | 39.4 | 42.3 | | 58.9 | | | | |
| Philakopi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 470 |
| Crete | 3.6 | | | 8.4 | | | | | | | | | | | | | | | | | | | | | | | 66.5 | | | |
| Haghia Triada | | | | | | | | | | | | | | | | | 24.3 | | | | | | | | | | | | | |
| Katsambas | | | | | 9.3 | | 9.8 | 10.5 | 11.4 | | | | | | | | | | | | | | | | | | | | | |
| Knossos | | 5.15 | | | | | | | | | | | 16 | | | | | | | 35 | | | | | | | | | | |
| Mavro Spelio | | | | | | | | | 11.4 | | | | | | | | | | | | | | | | | | | | | |
| Palaikastro | | | 7.8 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Tylissos | | | | | | 9.5 | | | | | | | | | | 23.9 | | | | | | | | | | | | | | |
| Cape Gelydonia | 3.6 | | | | 9.3 | 9.5 | | 10.5 | | | | | 16 | | | | | | | 35 | | 36 | | | | | 66.5 | | | 470 |
| Ebla | 3.6 | | 7.8 | 8.4 | | 9.5 | 9.8 | 10.5 | 11.4 | | 14.5 | | 16 | 16.3 | | | | 28.1 | | | | | 39.4 | | | | | | | |
| Kultepe | 3.6 | 5.15 | 7.8 | 8.4 | | | | | | 13.6 | 14.5 | | | | 20.2 | | | | | | | | | | | 58.9 | | | | |
| Nippur | | | | | | | | | | | | | | 16.3 | | | 24.3 | | | | | | | | | | | 84 | | |
| Ur | | | | | | | | | | | | | | 16.3 | | 23.9 | | | | | | | | 42.3 | | | | | 169.7 | |
| Chanhu-Daro | | | | | | | | | | | | | | | | | | | | | | | | | 54.5 | | | | | |
| Harappa | | | | | | | | | | | | 15 | | | | | | | 28.6 | | 35.8 | | 39.4 | | | | | | | |
| Mohenjo-Daro | | | | | | | | | | | | | | | | | | | 28.6 | | | | | | 54.5 | | | | | |

**Figure 1: Inter-regional weight measure matches among Fertile Crescent (green), Aegean (blue), including the island of Crete (dark blue), and the Indus Valley Civilization (pink). The green weights can be assumed to have spread from the Fertile Crescent, but the pink weights suggest a direct contact between the Aegean and the Indus Valley Civilization. Full Width Figures.**

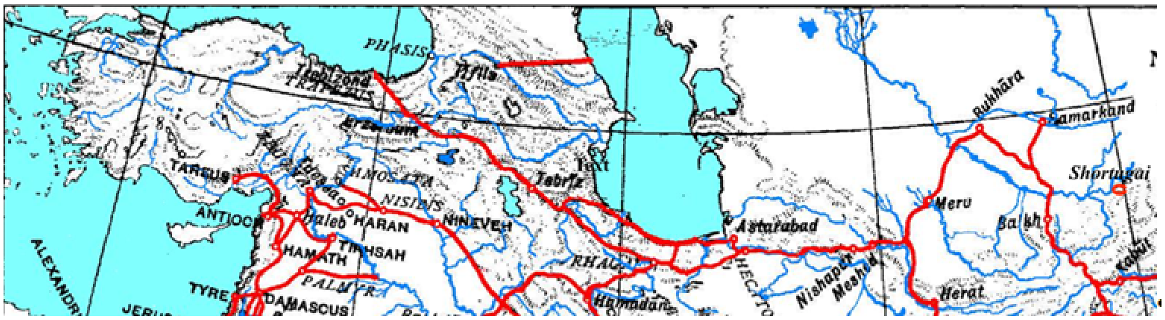


**Figure 2: The Trapezus-Tebriz-Astarabad-Meru-Buhhara-Shortugai trade route. (This map is based on https://en.wikipedia.org/wiki/File:C%2BB-Trade-Map1-HitherAsiaTradeRoutes.JPG)**

the Minoan civilization that avoids Mesopotamia. Hence, it seems important to describe some possible alternative trade routes that merchants may have taken in the Bronze Age between the Indus Valley and the Aegean area. This we will do in the next section.

## 4 DISCUSSION OF THE RESULTS

In Table 2, the matching weight groups highlighted in pink link the Indus Valley civilization with the Cycladic subgroup of the Minoan civilization. In particular, the direct Indus Valley-Minoan trade links seem to lead to Akrotiri, which was the main commercial center on the island Thera, which is now called Santorini, and Ayia Irini, which was the main commercial center on the island of Keos. This suggests the following possible alternate route.

### 4.1 An Alternative Route

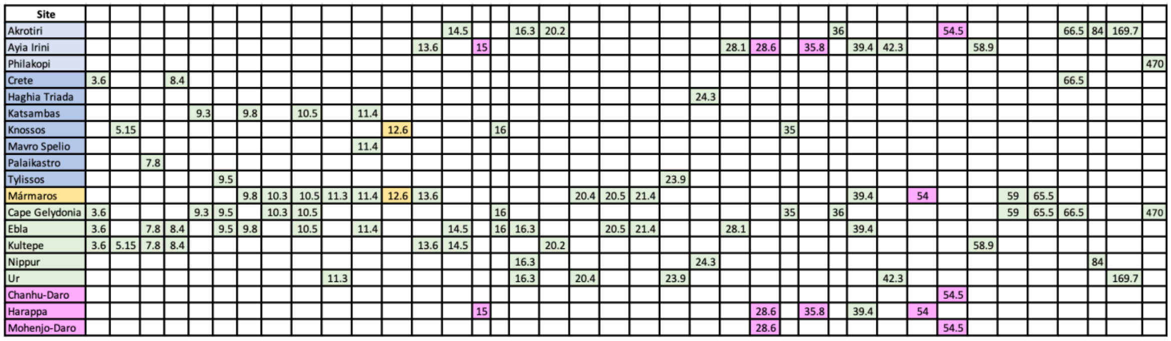
A possible route may have started from Shortugai, which was an Indus Valley civilization outpost town in the Himalayas. It is found today in Afghanistan. In Figure 2, we added Shortugai to a map that shows some ancient trade routes. It can be supposed that these routes already existed in the Bronze Age because we know that through Shortugai tin and lapis lazuli were transported to the Indus Valley civilization.

This alternative route would go from Shortugai west on the ancient Oxus River, which is now called the Amu Darya River. After Buhhara, a merchant could travel southwest to Meru, then to Astarabad, then to Tebriz, and finally to Trapezus. The Indus Valley and the Minoan traders could have met at Trapezus because the Minoans could sail through the Dardanelles and the Bosporus straits and sail along the northern coast of Turkey to reach Trapezus.

This scenario is plausible because the mountain dwelling Shortugai traders would need to travel through the mountains and the island dwelling Minoans would need to sail on the seas. An advantage of this direct Indus Valley-Minoan trade would be to circumvent the Mesopotamian intermediates, who would likely raise the price of the goods.

### 4.2 Further Evidence of a Northern Trade Route

If the Indus Valley traders would turn south around Lake Van, then they could also reach Cape Gelydonia and Ebla directly. There are many perfect matches between the Indus Valley weights and the Cape Gelydoani and Ebla weights without a matching Mesopotamian weight. This again suggests that the Indus Valley



| Site | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Akrotiri | | | | | | | | | | | | | | 14.5 | | | 16.3 | 20.2 | | | | | | | | | | 36 | | | | 54.5 | | | | 66.5 | 84 | 169.7 | |
| Ayia Irini | | | | | | | | | | | | | 13.6 | | 15 | | | | | | | | | 28.1 | 28.6 | | 35.8 | | 39.4 | 42.3 | | | 58.9 | | | | | | |
| Philakopi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 470 |
| Crete | 3.6 | | | 8.4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 66.5 | | | |
| Haghia Triada | | | | | | | | | | | | | | | | | | | | | | | 24.3 | | | | | | | | | | | | | | | | |
| Katsambas | | | | | 9.3 | | 9.8 | | 10.5 | | 11.4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Knossos | | 5.15 | | | | | | | | | | 12.6 | | | | 16 | | | | | | | | | | 35 | | | | | | | | | | | | | |
| Mavro Spelio | | | | | | | | | | | 11.4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Palaikastro | | | 7.8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Tylissos | | | | | | 9.5 | | | | | | | | | | | | | | | | 23.9 | | | | | | | | | | | | | | | | | |
| Mármaros | | | | | | | 9.8 | 10.3 | 10.5 | 11.3 | 11.4 | 12.6 | 13.6 | | | | | | 20.4 | 20.5 | 21.4 | | | | | | | | 39.4 | | 54 | | | 59 | 65.5 | | | | |
| Cape Gelydonia | 3.6 | | | | 9.3 | 9.5 | | 10.3 | 10.5 | | | | | | | 16 | | | | | | | | | | 35 | | 36 | | | | | | 59 | 65.5 | 66.5 | | | 470 |
| Ebla | 3.6 | | 7.8 | 8.4 | | 9.5 | 9.8 | | 10.5 | | 11.4 | | | 14.5 | | 16 | 16.3 | | | 20.5 | 21.4 | | | 28.1 | | | | | 39.4 | | | | | | | | | | |
| Kultepe | 3.6 | 5.15 | 7.8 | 8.4 | | | | | | | | | 13.6 | 14.5 | | | | 20.2 | | | | | | | | | | | | | | | 58.9 | | | | | | |
| Nippur | | | | | | | | | | | | | | | | | 16.3 | | | | | | 24.3 | | | | | | | | | | | | | | 84 | | |
| Ur | | | | | | | | | | 11.3 | | | | | | | 16.3 | | 20.4 | | | 23.9 | | | | | | | | 42.3 | | | | | | | | 169.7 | |
| Chanhu-Daro | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 54.5 | | | | | | | |
| Harappa | | | | | | | | | | | | | | | 15 | | | | | | | | | | 28.6 | | 35.8 | | 39.4 | | 54 | | | | | | | | |
| Mohenjo-Daro | | | | | | | | | | | | | | | | | | | | | | | | | 28.6 | | | | | | | 54.5 | | | | | | | |

**Figure 3: Inter-regional weight measure matches among Fertile Crescent (green), Aegean (blue), including the island of Crete (dark blue), the Indus Valley Civilization (pink), and Maramureș (orange).**

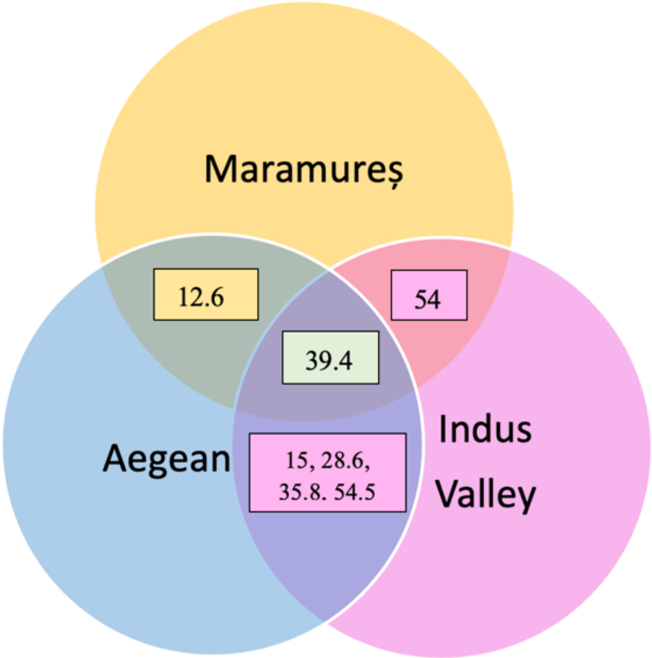


**Figure 4: A Venn diagram of those weights that are the most important indicators of direct trade. Pink weights indicate a direct trade between the Indus Valley civilization and either the Aegean or Maramureș. The orange weight indicates a direct trade between the Aegean and Maramureș. The green weight indicates a likely source from the Fertile Crescent.**

traders had direct contacts with traders from Cape Gelydonia and from Ebla.

Furthermore, there also may have been direct contacts between the Indus Valley civilization and some Bronze Age successors of the Old European culture in Southeastern Europe. In 1880, Hampel [3] reported a set of weights for the gold treasure found in today's county of Maramureș, Romania (formerly Mármaros, Hungary). Figure 3 shows the associations between those weights and the other sites from Ialongo et al. [4].

Figure 4 shows a Venn Diagram of those weights that indicate direct trade among any pair of the three periphery regions of the Aegean, the Indus Valley Civilization, and Maramureș. The 12.6 weight match between the Aegean and Maramureș suggests direct trade between the two regions. Similarly, the 54-weight match between the Indus Valley civilization and Maramureș suggests direct trade between those two regions. The 15, 28.6, 35.8, and 54.5 weight matches between the Aegean and the Indus Valley civilization implies direct trade between those two regions.

Figure 2 already suggested a trade route between the Black Sea port of Trapezus and the Indus Valley civilization town of Shortugai. From Trapezus one can sail on the Black Sea to the Danube Delta, and from there reach its tributaries, including the Tisza River, which leads to the Maramureș area as shown in Figure 5. This hypothetical route would be a natural connection between the Indus Valley

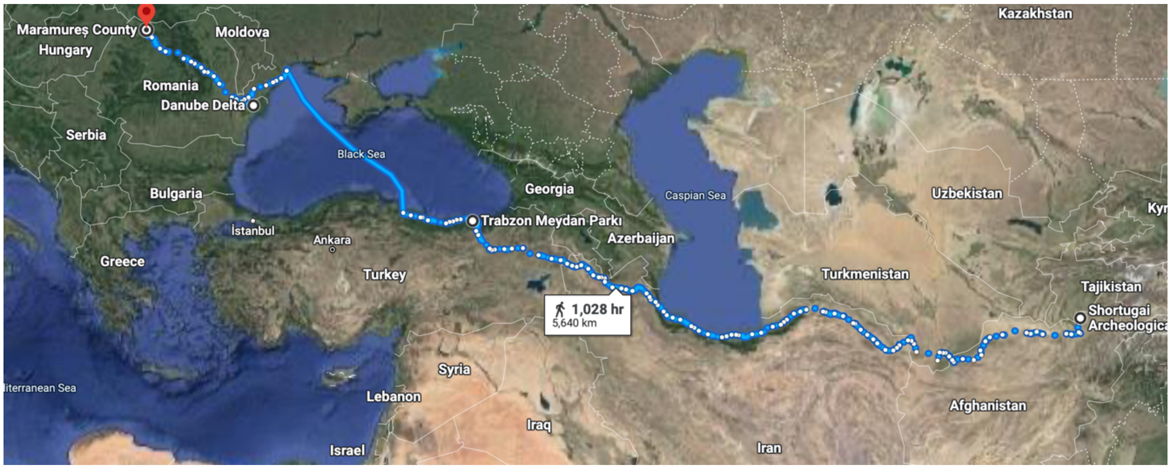


**Figure 5: A hypothetical Maramureș-Trapezus-Shortugai trade route. (This map is based on the direction finder of Google Maps https://www.google.com/maps using the keywords of Maramureș and Shortugai.)**

civilization and the Maramureș area. In addition, it is possible to sail from the Danube Delta to the Sea of Marmara and from there to the Aegean Sea. This hypothetical route would be the most suitable connection between Maramureș and the Minoan sites in the Aegean area. It was probably the Minoans who have sailed this sea route between the Danube Delta and the Aegean.

### 4.3 Related Work on the Minoan and the Indus Valley Civilizations

Our study of trade relations adds valuable information to the already known data regarding the Minoan and the Indus Valley civilizations. Recent advances in archaeogenetics yielded both mitochondrial and autosomal DNA data for the Minoans. Analyses of the mitochondrial [6] and the autosomal [10] DNA data consistently show that the Minoan society was composed of several groups. One group likely came from Anatolia, while the other group came from the Danube Basin and the western littoral area of the Black Sea [6]. The connection with the Anatolian farmers may go back to the earliest farmers in Crete because agriculture spread from Anatolia to the Aegean islands.

The connection with the Danube Basin may stem from the early Bronze Age. Many new migrants likely arrived at the island of Crete at the beginning of the Minoan civilization, which is called the Early Minoan period, around 3000 BC. Another wave of migrants arrived at the beginning of the Middle Minoan period around 2200 BC according to Arthur Evans. The exact chronology is debated by the archaeologists. However, they agree that writing was introduced to Crete during the Middle Minoan period.

Minoan writing had two forms: Cretan Hieroglyphs and the Linear A script. In 1991, Marija Gimbutas already pointed out some similarities between the Linear A script and the Danubian script signs. Her observation also suggests some population movement from the Danube Basin to Crete paralleling the archaeogenetic data. Hence, the earliest scribal class of the Middle Minoan period likely consisted of the new migrants from the Danube Basin, and the underlying language of the Linear A script could be related to the Pre-Indo-European language of the Old European civilization. The translation of twenty-eight mostly religious Linear A inscriptions suggests that the scribal language was a Uralic language [5]. The Uralic language speaking peoples were assumed to have had a homeland somewhere near the Ural Mountains. However, it is possible that this language family goes back to the Mesolithic or Paleolithic period because there are no cognate agriculture-related words between the Finno-Permic and the Ugric branch of languages. In that early period, the Uralic homeland was likely in the Danube Basin rather than anywhere more to the north.

Surprisingly, there are many cognate Pre-Greek and Ugric words [8]. These Pre-Greek cognate words likely were borrowed from the Minoan language by Greek. In addition, a graph-based algorithmic analysis of Minoan inscriptions was able to show that the Minoan language had front-back vowel harmony [9]. Front-back vowel harmony likely was already present in the Proto-Uralic language and is wide-spread within the Uralic language family. Front-back vowel harmony is not a characteristic of Indo-European languages.

The scribal language may have been different from the common Minoan language because the Minoan civilization may have been multilingual during the Middle Minoan period. Homer (Odyssey, book 19, lines 172-177) described the island of Crete as a multilingual and multiethnic place around 800 BC. While Homer mentions Achaeans and Dorians, two well-known Greek groups, he also mentions other groups that could be Pre-Greek: the Eteocretans, or ‘true Cretans’, the Kydones, and the Pelasgians. Given that the Mycenaean civilization followed the Minoan civilization and Crete remained continuously under Greek dominance until Homer, it is hard to explain the presence of these apparently Pre-Greek groups unless they were already in Crete during the Minoan period.

A detailed art motif analysis [7] also identified three sets of Minoan art motifs. The first set contains art motifs that spread from the Near East via the spread of agriculture. These motifs spread both eastward and westward together with agriculture and can be found in both the Indus Valley and Crete during the Early Minoan period. The second set of motifs apparently originated in the Danube Basin because they first appear there in the Neolithic or early Bronze Age. This second set of motifs first appear on Crete during the Middle Minoan period. Finally, the third set contains art motifs that were likely brought in by the Mycenaeans because they appear on Crete during the Late Minoan period, when the island was already occupied by the Mycenaeans.

The Indus Valley script has a close connection with the Sumerian pictograms but only a distant relation with the Linear A and Linear B scripts [2]. That is likely because syllabic writing developed only in the Bronze Age.

## 5 CONCLUSIONS AND FUTURE WORK

We hope these techniques will enable the discovery of other possible trade routes among ancient civilizations. In general, the method could be applied to discover other relations too that rely on copying some other metric such as length or volume. Our study of trade relations adds valuable information to the already known similarities. Each type of similarity shown serves as a piece of a grand mosaic that depicts strong and vibrant connections among Bronze Age cultures that were previously viewed as isolates.

## REFERENCES

[1] Beekes, R. S. P. 2009. Etymological Dictionary of Greek. Brill NV. Leiden, Netherlands.

[2] Shruti Daggumati and Peter Z. Revesz. 2021. A method of identifying allographs in undeciphered scripts and its application to the Indus Valley Script. Humanities and Social Sciences Communications, 8, 50. https://doi.org/10.1057/s41599-021-00713-0

[3] József Hampel, 1880. Marmarosmegyei aranylelet. Archaeologiai Közlemények, 14, 29-32.

[4] Nicola Ialongo, Raphael Hermann and Lorenz Rahmstorf. 2021. Bronze Age weight systems as a measure of market integration in Western Eurasia, Proceedings of the National Academy of Sciences, 118, 27. https://doi.org/10.1073/pnas.2105873118

[5] Peter Z. Revesz, 2017. Establishing the West-Ugric language family with Minoan, Hattic and Hungarian by a decipherment of Linear A. WSEAS Transactions on Information Science and Applications, 14, 306-335.

[6] Peter Z. Revesz, 2019. Minoan archaeogenetic data mining reveals Danube Basin and western Black Sea littoral origin. International Journal of Biology and Biomedical Engineering, 13, 108-120.

[7] Peter Z. Revesz, 2019. Art motif similarity measure analysis: Fertile Crescent, Old European, Scythian and Hungarian elements in Minoan culture. WSEAS Transactions on Mathematics, 18, 264-287.

[8] Peter Z. Revesz, 2020. Minoan and Finno-Ugric regular sound changes discovered by data mining. Proceedings of the 24th International Conference on Circuits, Systems, Communications and Computers, IEEE Press, 241-246.

[9] Peter Z. Revesz, 2020. A vowel harmony testing algorithm to aid in ancient script decipherment. Proceedings of the 24th International Conference on Circuits, Systems, Communications and Computers, IEEE Press, 35-38.

[10] Peter Z. Revesz. 2021. Data mining autosomal archaeogenetic data to determine Minoan origins. In Proceedings of the 25th International Database Engineering and Applications Symposium (IDEAS'21) ACM Press, New York, NY, 46-55. https://doi.org/10.1145/3472163.3472178

[11] Evgenia Tsafou, and Juan José García-Granero, 2021. Beyond staple crops: Exploring the use of 'invisible' plant ingredients in Minoan cuisine through starch grain analysis on ceramic vessels. Archaeological and Anthropological Sciences, 13, 128. https://doi.org/10.1007/s12520-021-01375-4

[12] Bernardo Urbani and Dionisios Youlatos, 2020. A New Look at the Minoan 'Blue' Monkeys. Antiquity, 94, 374. https://doi.org/10.15184/aqy.2020.29

[13] Naama Yahalom-Mack, Daniel M. Finn, Yigal Erel, Ofir Tirosh, Ehud Galili and Assaf Yasur-Landau, 2022. Incised Late Bronze Age lead ingots from the southern anchorage of Caesarea, Journal of Archaeological Science: Reports, 41. https://doi.org/10.1016/j.jasrep.2021.103321

# Decision making with Clustered Majority Judgment

Arianna Anniciello
University of Napoli Federico II
Napoli, Italy, EU

Emanuele d'Ajello
University of Napoli Federico II
Napoli, Italy, EU
emanuele.dajello@gmail.com

Davide Formica
Copernicani
Milano, Italy, EU
frmdvd@gmail.com

Elio Masciari
University of Napoli Federico II
Napoli, Italy, EU
elio.masciari@unina.it

Gaia Mattia
University of Napoli Federico II
Napoli, Italy, EU
ga.mattia@studenti.unina.it

Stefano Quintarelli
Copernicani
Milano, Italy, EU
stefano@quintarelli.it

Davide Zaccarella
University of Napoli Federico II
Napoli, Italy, EU
d.zaccarella@studenti.unina.it

## ABSTRACT

Making decisions quickly and efficiently is essential in all areas of existence; when the decision concerns a fair number of voters and options to vote, it is sometimes appropriate to choose a voting system that represents the voting population well, without excluding the preferences of minorities.

In this paper we present a voting system that aims to be an enhancement of *Majority Judgment* through unsupervised machine learning techniques, in particular the cluster and, in addition, a criterion for obtaining a multiwinner result has also been added. After exposing its functioning, a case study is presented to test its applicability, which leads to multiple fields of interest and it is not limited exclusively to purely political occasions.

## CCS CONCEPTS

• **Data Science and applications**;

## KEYWORDS

Decision Making, Social Choice, Cluster, Majority Judgement, K-Medoids

**ACM Reference Format:**
Arianna Anniciello, Emanuele d'Ajello, Davide Formica, Elio Masciari, Gaia Mattia, Stefano Quintarelli, and Davide Zaccarella. 2022. Decision making with Clustered Majority Judgment. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3548785.3548798

## 1 INTRODUCTION

In this section we deal with general behiavour of some voting rules, like the majority and the premised rule, trying to highlight pros and limits. Then, we're interested in finding a model which guarantees inclusion for minorities during multiwinner decision-making process.

A more 'inclusive' voting rule leads to implement a clustered version of the chosen model (*Majority Judgement*): clusters are created taking into account the similarity between the expressed preferences; for each of them, *Majority Judgement* rule is applied to return a ranking over the set of candidates. Now we explore differences provided by different voting rules. Using a specific voting rule determines limitations and advantages: one could choose a voting rule in order to avoid a tactical voting approach, renouncing on judgements' representiveness. We consider three agents who express their judgements ("Yes" or "No") for four statements $A$, $B$, $A \wedge B$ and $A \longleftrightarrow B$, and compare outcomes from majority and premised-based rule. The latter take majority decisions on A and B and then infers conclusions on the other two propositions.

As shown in the table 1, results are different considering the used role.

We now focus on Agent 2 case: he's represented in just one of the propositions (A), and in the remaining cases his judgement doesn't agree with the outcome. So, Agent 2 could think about manipulating the final result, by pretending a disagreement for A. As consequence, the premised model reacts by providing as final outcome on 3 agents' votation a "No" for both $A \wedge B$ and $A \longleftrightarrow B$, as originally expressed by Agent 2.

| | A | B | $A \wedge B$ | $A \longleftrightarrow B$ |
|---|---|---|---|---|
| Agent 1 | Yes | Yes | Yes | Yes |
| Agent 2 | Yes | No | No | No |
| Agent 3 | No | Yes | No | No |
| **Premised rule** | Yes | Yes | Yes | Yes |
| **Majority** | Yes | Yes | No | No |

**Table 1: Three agent case of voting**

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548798

In such a way, strategic voting is not avoided for Agent 2. This is the major drawback of using premised voting as rule.
On the other hand, a paradoxal aspect arises considering the majority rule: outcomes of the latest two propositions are not consistent with "Yes" value assigned to both A and B.
This is known as *discursive dilemma*, the inconsistency problem in judgement aggregation based on *majority rule* [11].
Results can be interpreted as follows: in case of majority rule, it is always in Agent's interests to give his true preference. For this reason we consider majority rule as a transparent (with no tactical voting) asset in decisional process, while trying to deal with its intrinsic issues due to judgement aggregation [13].
Our aim is not solving above-mentioned dilemma, but using majority rule as a baseline for a more refined model (*Majority Judgement*), with the use of clusters for a more inclusive rule.

## 2 STRATEGIES OF DECISION MAKING

### 2.1 Collective decision process and Majority Judgement

During business meetings it is sometimes difficult to bring together the ideas of all participants regarding important decisions: this can lead to slowness and non-productivity. In many contexts, to resolve this type of situation, there is a majority vote. This is a good method but it is a rough approximation of the collective desire, excluding part of those who represent the minority. In order to limit this effect, an algorithm has been developed starting from the traditional *Majority Judgement* (MJ), enhanced with a clustering technique which, before calculating the resulting winner, divides the voting population into groups with similar preferences.
Whereas choice theory is concerned with individuals making choices based on their preferences, social choice theory is concerned with how to translate the preferences of individuals into the preferences of a group, the case in which this process is revealed is the one which concerns the electoral vote [6]. During voting, electors in a democracy choose one candidate among a list of many candidates, while in a jury decision the individual judges evaluate competitors in a competition, ranking them.
Arrow's impossibility theorem in social choice theory states that when voters have three or more distinct alternatives (options), no ranked voting electoral system can convert the ranked preferences of individuals into a community-wide (complete and transitive) ranking while also meeting the specified set of criteria: unrestricted domain, non-dictatorship, Pareto efficiency, and independence of irrelevant alternatives [1]. In [18], Condorcet and Borda methods and limits, Arrow's impossibility theorem and MJ are illustrated and the results of the general elections have shown that voting systems can run into the Arrow's paradox. A known example is the 2000 US presidential election where the presence of a minor candidate, Ralph Nader, who had no chance of winning, made Bush the winner in Gore's place. It's legit to suppose that the presence of Nader made Gore's votes dispersed, given their political positions. Moreover, Nader supporters preferred Gore to Bush. However, the American voting system permits a single vote in a single round where all the candidates are available options, and the one with most votes is the winner. In such a way, voters haven't fully expressed their preferences: the winner turned out to be just the candidate more resilient to Arrow's paradox, and not the most preferred one.
So, MJ is a voting technique proposed by two mathematicians in 2007, Michel Balinski and Rida Laraki, aiming to overcome traditional voting methods' paradoxes and inconsistencies. In [3], Balinski and Laraki present MJ, a model that overcomes traditional models which are prone to suffer from incoherence, impossibility and incompatibility: in[5] authors reject the assumption of traditional methods that electors don't really make a personal ranking of candidates and highlight this false belief as the reason behind the inadequacy of voting models different from MJ.
In[4] they also present the case of the French presidential elections of 2002 as another example of Arrow's paradox: the winner depends on the presence or absence of candidates, including those who have absolutely no chance of winning. Balinski and Laraki's point is that only the presence of a common language leads to a coherent collective decision, and consequently a greater expressiveness in the voting system minimizes the paradoxal effects. MJ makes it possible: it asks for electors/judges to express a judgment on all the candidates/competitors, using a known common language. Each quality is associated with a numeric score and the candidate with the highest median score is the winner. In case of tie, a tiebreaker is used which considers how "broad" that median grade is. Even though it's not possible to avoid completely strategical voting, MJ strongly resists manipulation. And this is one of the features we wanted to consider when modelling an inclusive and transparent voting rule.

### 2.2 Social theory's requirements

May (1952) [15] introduced four such requirements for majority voting rule must satisfies:[7]

- **Universal domain**: the domain of admissible inputs of the aggregation rule consists of all logically possible profiles of votes $< v_1, v_2, ..., v_n >$, where each $v_i \in [-1, 1]$ (to cope with any level of 'pluralism' in its inputs);
- **Anonimity**: applying any kind of permutation on individual preferences does not affect the outcome (to treat all voters equally), i.e.,
$$f(v_1, v_2, ..., v_n) = f(w_1, w_2, ..., w_n) \tag{1}$$
- **Neutrality**: each alternative has the same weight and for any admissible profile $< v-1, v_2, ..., v_n >$, if the votes for the two alternatives are reversed, the social decision is reversed too (to treat all alternatives equally), i.e.
$$f(-v_1, -v_2, ..., -v_n) = -f(v_1, v_2, ..., v_n) \tag{2}$$
- **Positive responsiveness**: For any admissible profile $< v_1, v_2, ..., v_n >$, if some voters change their votes in favour of one alternative (say the first) and all other votes remain the same, the social decision does not change in the opposite direction; if the social decision was a tie prior to the change, the tie is broken in the direction of the change, i.e., if $w_i > v_i$ for some $i$ and $w_j = v_j$ for all other $j$] and $f(v_1, v_2, ..., v_n) = 0$ or 1, then $f(w_1, w_2, ..., w_n) = 1$.

A multi-winner election $(V, C, F, k)$ is defined by a set of voters $V$ expressing preferences over a number of candidates $C$, and then a voting rule $F$ returns a subset of size $k$ winning candidates. A voting rule can pact on different types of ordered preferences, even though the most common have a pre-fixed linear order on the alternatives. In most of cases, these are chosen *a priori.*

Formally we denote set of judgements performed by the i-th voter as profile preferences $\mathrm{P}_i$. Each profile contains information about the grade of candidates by voters. The voting rule $F$ associates with every profile $P$ a non-empty subset of winning candidates.

In multi-winner elections more precise traits are required, compared to the ones stated in May's theory [9]. Indeed:

- **Representation**: for each subset of voters

$$V_i \in V \text{ (with } |V_i| \geq \left\lfloor \frac{n}{k} \right\rfloor) \tag{3}$$

  at least one successful candidate is elected from that partition;
- **Proportionality**: for each subset of voters

$$V_i \in V \text{ (with } |V_i| \geq \left\lfloor \frac{n}{k} \right\rfloor) \tag{4}$$

  number of elected candidate is proportional to the subset's size.

An implicit assumption so far has been that preferences are ordinal: preference orderings contain no information about each individual's strength or about how to compare individuals' preferences with one another. In voting contexts, this assumption may be acceptable, but in welfare-evaluation contexts - when a social planner seeks to rank different social alternatives in an order of social welfare - the usage of further information may be justified.

### 2.3 Single-winner Majority judgement

In order to describe the majority judgement, we need to use a table that refers to ranking for all the candidates $C$, by using tuples [2]. Suppose having six possible choices we may use the words: *excellent, very good, good, discrete, bad, very bad.*

So each candidate is described by a bounded set of vote.

Winner is found comparing recursively median grade between candidates: first, grades are ordered in columns from the highest to the lowest according to the order relation, then the middle column (lower middle if number of grades are even) with the highest grade between candidates'row is selected. If there's a tie, algorithm keeps on discarding grades equal in value to the shared median, until one of the tied candidate is found to have the highest median. Our aim is to generalize this single-winner strategy to a multi-winner one, using a clustering approach in judgement aggregation. So, we first discuss our choices for clusters and then we describe the algorithm.

## 3 CLUSTERS

### 3.1 Categories of clusters

Different types of cluster share the ability to group data with some common features. Some of most important types are:

1. **Connectivity models**: similarity or differences arise from the distance made between data points.

Two possible approaches are: *bottom-up* where each data point is a cluster and then pairs of clusters are merged; *top-down*, where observations are included in one cluster and then it's segregated; limitations of this model are shown because of impossibility to make changes to an already craeted cluster;

2. **Distribution models**: probabilities about belonging to a particular distribution once the cluster is created are computed. Limitations are shown where no precise constraints are given, as this model tends to overfit data;

3. **Density models**: cluster are created in areas of higher density of data points, while the remaining can be grouped into an arbitrary and distribution-less shaped area; this features make the model likely to be less sensitive to noise than other types of clusters;

In our case, clusters need to satisfy social theory's requirements that determines a pretty fixed structure, but with no assumption about data distribution. For these reasons, we focus on a different class of clustering algorithm, the *centroid models.*

### 3.2 K-Medoids

For our goal, namely selecting winners from a group of candidates, *K-medoids* clustering is used. Our choice is due to the fact that averaging methods like K-means clustering could result in a solution which doesn't belong to the candidate list. In our case, medoid is a data point (unlike the centroid) which has the least total distance to the other members of its cluster [10].

Another advantage for this choice is that the mean of the data points is a measure that gets highly affected by the extreme points; so, in K-Means algorithm, the centroid may get shifted to a wrong position and hence result in incorrect clustering if the data has outliers. On the contrary, the K-Medoids algorithm is the most central element of the cluster, such that its distance from other points is minimum. Thus, compared to K-Means algorithm, K-Medoids is more robust to outliers and noise. [8].

The used K-medoid algorithm is in the python `sklearn` library [17]. This library supports *partitioning around medoids* (PAM) [12] proposed by Kaufman and Rousseeuw (1990). The workflow of PAM is described below [16].

The PAM procedure consists of two phases: *BUILD* and *SWAP*:

- In the BUILD phase, primary clustering is performed, during which $k$ objects are successively selected as medoids.
- The SWAP phase is an iterative process in which the algorithm makes attempts to improve some of the medoids. At each iteration of the algorithm, a pair is selected (medoid and non-medoid) such that replacing the medoid with a non-medoid object gives the best value of the objective function (the sum of the distances from each object to the nearest medoid). As there is the possibility of improving the outcome of the objective function, the procedure is repeated.

Suppose that $n$ objects having $p$ variables each should be grouped into $k$ ($k < n$) clusters, where $k$ is known. Let us define j-th variable of object $i$ as $X_{ij}$ ($i = 1, ..., n; j = 1, ..., p$). As a dissimilarity measure is used the Euclidean distance, that is defined, between object $i$ and object $j$, by:

$$d_{ij} = \sqrt{\sum_{a=1}^{p} (X_{ia} - X_{ja})^2} \tag{5}$$

where $i$ and $j$ range from 1 to $n$. The medoids is selected in this way:

- calculate the Euclidean distance between every pair of all objects;
- calculate $v_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{n} d_{il}}$;
- sort all $v_j$ for $j = 1, ..., n$ in ascending order and select the first $k$ object that have smallest initial medoids value;
- from each object to the nearest medoid we can obtain the initial cluster result;
- calculate the sum of distances from all objects to their medoids;
- update the current medoid in each cluster by replacing with the new medoid, selected minimizing the total distance from a certain object to other objects in its cluster;
- assign each object to the nearest medoid and obtain the cluster result;
- calculate the sum of distance from all objects to their medoids, so if the sum is equal to the previous one, then stop the algorithm; otherwise, go back to the update step.

In our case, prior knowledge about the number of winners is required, and identified clusters are restricted in minimum size that is number of voters on the number of candidates ($\frac{n}{k}$).

### 3.3 Clustered Majority Judgement

For each cluster majority judgement is applied, and a final ranking of candidates is returned [14]. Given $k$ the number of candidates to be elected, algorithm seeks the optimal number of cluster to create. The number of clusters ranges from 1 to $k$ and has to satisfy an additional requirement: if a tie occurs and *k'* vacant seats are left, algorithm is repeated *k'* times until tie's broken. In case there's no broken tie, the number of cluster is changed.
We present the relevant steps in pseudocode:

(1) set the number of winners as maximum number of clusters;
(2) cluster are created decreasing the maximum number of clusters until the optimal number is achieved. This number is bound by the size of cluster, that satisfies the following equation: *number of voters : number of winners = number of voters in one cluster : one winner*;
(3) the function *winners* calculates the median for every cluster is created;
(4) check that winners from clusters are different between each other ; in case these are not distinct (condition="ko" on pseudocode) algorithm goes back to step 2 with a maximum number of cluster equal to the number of vacant seats and the successive steps are executed until all seats have been filled.

**Algorithm 1**

```
Require: $k \geq 0$
Ensure: $n\_winners = (n_1, ..., n_k), k > 1$
  $k \leftarrow number\_winners$
  $max\_cluster \leftarrow k$
  $condition \leftarrow "ko"$
  while $condition = "ko"$ do
      $cluster\_list \leftarrow cluster(vote\_list)$
      for all list_cluster do
          $winners\_per\_cluster \leftarrow compute\_winners(cluster)$
          $all\_winners \leftarrow list\_of\_all\_winners(winners\_per\_cluster)$
      end for
      $list\_winner\_distinct = list\_of\_all\_distinct\_winners(all\_winners)$
      $option\_remaining \leftarrow number\_winners - len(list\_winner\_distinct)$
      if $option\_remaining = 0$ then
          $condition =' ok'$
      else
          $k \leftarrow option\_remaining$
          $condition \leftarrow' ko'$
      end if
  end while
```

### 3.4 Case study: Using clustered majority judgement to maximize agreement

A local cultural association organizes a themed film club annually during the summer. It is possible both to access the single event and to make a subscription. To maximize the sale of season tickets, a portion of associated users responded to a survey concerning the numerous possible topics of the film club. The survey was structured so that each topic is assigned a value from 1 to 7 by the voters, that corrispond to *Excellent, Very Good, Good, Acceptable, Poor, To Reject, No Opinion*, in order to implement an interesting comparisons of Majority Judgement (MJ) and Clustered Majority Judgement (CMJ). A new input parameter of CMJ - as compared to MJ - is introduced: the number of winners, that is fixed equal to 2. Into this experiment 37 voters took part and the algorithm form two clusters, exactly like the number of winners.

| Cluster | Cluster size | Winner |
|---|---|---|
| Cluster 1 | 21 | Satire |
| Cluster 2 | 16 | Science fiction |

**Table 2: CMJ results**

| Ranking MJ | Candidate |
|---|---|
| 1 | Satire |
| 2 | History |

**Table 3: Top 2 of single-winner Majority Judgement applied to voters**

We can compare CMJ results with single-winner MJ ranking, comparing Table 3 3 and Table 2 2. We notice both for Majority Judgement and Clustered Majority Judgement the tendency to avoid the favourite topic, focusing on the moderate ones. Furthermore, in CMJ case, the expressed judgements are quite polarizing and the two formed cluster seems in opposition between each other, because the preferred topic for one are tendentially negatively judged by the other one.
In case of MJ, the solution is *Satire* and *History*, so, considering only the first two classified, seeing that themes tend to be similar, it is

likely that a large part of the subscriptions is purchased exclusively by people grouped within a single cluster, which is the largest one. We see instead that the CMJ, in addition to *Satire* topic, also considers the preponderant preference of the second cluster. For this reason, this strategy is the most likely to maximize the subscription purchases.

## CONCLUSIONS

In section 1, we explored different voting rules and their limitations. In section 2, a more fined model of majority rule, Majority Judgement, has been presented as the best model to avoid the incoherence produced by traditional voting methods .

Then we present our generalization of Majority Judgement as a multi-winner strategy, thanks to the use of clusters. After that, a case study is reported, with a particular attention to the comparison between MJ and CMJ results.

The CMJ, as shown, represents the optimal compromise in case of polarized groups (clusters). This appears to be the best model in order to maximize agreement, as shown in the case study, as it represents well the minorities, by taking into account their preferences more carefully than a simple Majority Judgement model.

In spite of non-deterministic nature of K-Medoids, Clustered Majority Judgement is thought to be used in high populated disputes. For these reasons, we feel confident about clustering's role of considering all different perspectives could be shown in these situations.

As this implementation depends only on some fixed parameters, like the grades, their number and the number of winners to select, the algorithm can find space in any type of decision making process.

## REFERENCES

[1] K.J. Arrow, Y.C.D.E.C.F.R. Economics, C. economics, K.W.H.C. Collection, Y. University, Y.U.C.F.R. Economics, and C.F.R.E.Y. University. 1963. *Social Choice and Individual Values.* Number No. 12 in Cowles Foundation Monographs Series. Wiley. https://books.google.it/books?id=Lo2uCECV__8C

[2] Michel Balinski. 2008. Fair Majority Voting (or How to Eliminate Gerrymandering). *Am. Math. Mon.* 115, 2 (2008), 97–113. http://www.jstor.org/stable/27642416

[3] Michel Balinski and Rida Laraki. 2007. A Theory of Measuring, Electing and Ranking. *Proceedings of the National Academy of Sciences of the United States of America* 104 (06 2007), 8720–5. https://doi.org/10.1073/pnas.0702634104

[4] Michel Balinski and Rida Laraki. 2011. *Election by Majority Judgment: Experimental Evidence.* Springer New York, New York, NY, 13–54. https://doi.org/10.1007/978-1-4419-7539-3_2

[5] Michel Balinski and Rida Laraki. 2014. Judge: Don't Vote! *Oper. Res.* 62, 3 (2014), 483–511. https://doi.org/10.1287/opre.2014.1269

[6] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). 2016. *Handbook of Computational Social Choice.* Cambridge University Press. https://doi.org/10.1017/CBO9781107446984

[7] Luciano Caroprese and Ester Zumpano. 2020. Declarative Semantics for P2P Data Management System. *J. Data Semant.* 9, 4 (2020), 101–122. https://doi.org/10.1007/s13740-020-00115-6

[8] Michelangelo Ceci, Roberto Corizzo, Fabio Fumarola, Michele Ianni, Donato Malerba, Gaspare Maria, Elio Masciari, Marco Oliverio, and Aleksandra Rashkovska. 2015. Big Data Techniques For Supporting Accurate Predictions of Energy Production From Renewable Sources. In *Proceedings of the 19th International Database Engineering & Applications Symposium, Yokohama, Japan, July 13-15, 2015*, Bipin C. Desai and Motomichi Toyama (Eds.). ACM, 62–71. https://doi.org/10.1145/2790755.2790762

[9] Adrien Fabre. 2021. Tie-breaking the highest median: alternatives to the majority judgment. *Soc. Choice Welf.* 56, 1 (2021), 101–124. https://doi.org/10.1007/s00355-020-01269-9

[10] Bettina Fazzinga, Sergio Flesca, Filippo Furfaro, and Elio Masciari. 2013. RFID-data compression for supporting aggregate queries. *ACM Trans. Database Syst.* 38, 2 (2013), 11. https://doi.org/10.1145/2487259.2487263

[11] A. Subramoney G. Bellec, F. Scherr. 2020. A solution to the learning dilemma for recurrent networks of spiking neurons. In *Nat Commun.*

[12] Leonard Kaufman and Peter J. Rousseeuw. 2008. *Partitioning Around Medoids (Program PAM).* John Wiley & Sons, Inc., 68–125. https://doi.org/10.1002/9780470316801.ch2

[13] Jon M. Kleinberg. 2002. An Impossibility Theorem for Clustering. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, Suzanna Becker, Sebastian Thrun, and Klaus Obermayer (Eds.). MIT Press, 446–453. https://proceedings.neurips.cc/paper/2002/hash/43e4e6a6f341e00671e123714de019a8-Abstract.html

[14] Andrea Loreggia, Nicholas Mattei, and Stefano Quintarelli. 2020. Artificial Intelligence Research for Fighting Political Polarisation: A Research Agenda. In *Proceedings of the First International Forum on Digital and Democracy. Towards A Sustainable Evolution 2020, Venice, Italy, December 10-11, 2020 (CEUR Workshop Proceedings, Vol. 2781)*, Patrizia Feletig, Andrea Loreggia, Andrea Resca, and Stefano Quintarelli (Eds.). CEUR-WS.org, 24–33. http://ceur-ws.org/Vol-2781/paper2.pdf

[15] Kenneth O. May. 1952. A set of indipendent necessary and sufficient conditions for simple majority decision. In *Carleton College.*

[16] Hae-Sang Park and Chi-Hyuck Jun. 2009. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* 36, 2 (2009), 3336–3341. https://doi.org/10.1016/j.eswa.2008.01.039

[17] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2012. Scikit-learn: Machine Learning in Python. *CoRR* abs/1201.0490 (2012). arXiv:1201.0490 http://arxiv.org/abs/1201.0490

[18] P. Serafini. 2019. *La Matematica in Soccorso Della Democrazia: Cosa Significa Votare e Come Si Può Migliorare il Voto.* Independently Published. https://books.google.it/books?id=uX6ixAEACAAJ

# Quality prediction in a smart factory: a real case study

Sana Ben Abdallah Ben Lamine
Riadi Laboratory ENSI/ISAMM
University of Manouba
Manouba, Tunisia
sana.benabdallah@riadi.rnu.tn

Malek Kamoua
Riadi Laboratory ENSI/ISAMM
University of Manouba
Manouba, Tunisia
kamoua.malek@isamm.u-manouba.tn

Haythem Grioui
Addixo
Montigny-le-Bretonneux, France
haythem.grioui@addixo.com

## ABSTRACT

The Industry 4.0 concept refers to new production patterns that include new technologies, manufacturing elements, and workforce organizations. It creates highly efficient production systems that change production processes, reduce production costs and improve product quality. Quality 4.0 is an evolution of Industry 4.0, which is a modification of traditional quality control charts. In this paper, our motivation is to improve manufacturing processes as we monitor product's quality by improving the percentage of correctly manufactured products thereby achieving efficiency. A four-layer decision-making architecture is proposed where different models and techniques are applied and a comparative study is achieved on real industrial case study: 1) data exploration layer, 2) feature engineering layer, 3) modeling layer, in which three categories of time series forecasting algorithms are experimented: statistical model (ARIMA), machine learning models (Random forest and XGBOOST) and deep learning models (Stacked LSTM and Transformer-based model), and finally 4) interpretation layer. The transformer-based model scored the best. With the classification model's interpretation, we deducted the recommended values to monitor the product's quality in order to reach relatively zero defects.

## CCS CONCEPTS

• **Information systems → Data analytics**;

## KEYWORDS

Smart factory, Industry 4.0, Quality 4.0, time series forecasting, classification

**ACM Reference Format:**
Sana Ben Abdallah Ben Lamine, Malek Kamoua, and Haythem Grioui. 2022. Quality prediction in a smart factory: a real case study. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3548785.3548796

## 1 INTRODUCTION

Industry 4.0 makes full use of emerging technologies and the rapid development of machines and tools to improve the level of industry. As a result, the manufactured product will be of better quality and production systems will be more efficient and easier to maintain. According to [1], a survey conducted in 2019, 32 % of companies surveyed in Indonesia experienced a 31 to 50 % improvement in operational productivity and performance when adopting industry 4.0. Quality 4.0 as leveraging traditional quality control techniques gained through the latest technology to deliver new levels of excellence at the functional and operational levels [2]. Quality 4.0 is an evolution of Industry 4.0, which is a modification of traditional quality control charts. Organizations can monitor processes and extract data from real-time sensors. Quality 4.0 is an evolution of Industry 4.0, which is a modification of traditional quality control charts using the latest technology to deliver new levels of excellence at the functional and operational levels [2]. Manufacturers applying Quality 4.0 technology have achieved remarkable efficiency in quality management, thereby expanding market share, promoting innovation, and improving their ability to face challenges and enhance brand recognition [2]. In this context we propose a four-layer decision-making architecture where different AI models and techniques are applied and a comparative study is achieved. This proposal aims at forecasting real world manufactured product's properties in order to enhance its quality. In fact, the forecasted values are classified and interpreted in order to give recommended values to monitor the manufactured product's quality. The problem of hyperparameter optimisation resides in the choice of the best set of hyperparameters for the learning algorithm using several techniques such as grid search and random search. According to [26], random search is the best method of parameter search for small dimensions. For our study, we tested the following common hyperparameters: Batch_size, epochs, optimizer and dropdown. For each model we add other specific parameters. For this task, we used Weights & Biases, a machine learning platform to track our experiments, versions and evaluate our models' performance. Structure of the paper is as follows. Section 2 provides related work of AI in industry 4.0 along side with studies carried out in the context of time series forecasting in an industrial environment. In section 3 we introduce our proposal. Sections 4 is dedicated to experimentation and results. Finally, section 5 concludes the paper and announces future works.

## 2 RELATED WORKS

Zhang et al [8], proposed in 2017 an approach to monitor the performance degradation of dynamic systems in the context of manufacturing. This approach is based on LSTM to characterise the degradation behaviour of the system and then predict the remaining useful life (RUL). The long-term dependent properties embedded in the LSTM framework are intended to capture the interrelationships of the time series of data measured by the monitored system, allowing for more accurate predictions of future behaviour. The experiments

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*

ACM ISBN 978-1-4503-9709-4/22/08.
https://doi.org/10.1145/3548785.3548796

consist in comparing different models. The authors obtained the following results: For SVR: rmse = 20.96, MLP rmse= 20.84, LSTM rmse=18.07, Bidirectional-LSTM rmse=15.42. Bidirectional LSTM networks perform better. Futterer et al [9], proposed in 2017 an application prospects of several supervised learning methods for time series classification in BACS (Building Automation and Control Systems) as they trained thirteen types of classifiers including complex tree, average tree, simple tree, linear support vector machine, KNN subspace, augmented RUS tree, good KNN, raw KNN and random forests. Bagged trees scored the highest demonstrated average classification accuracy (56.76%), with the maximum accuracy level of 76.54%. However, the maximum accuracy achieved by random forests was even higher, reaching 78.95%. An extensive part of this literature uses traditional machine learning algorithms [9] [5] [7] such as random forest and linear regression. In recent years, an important number of research papers revolve around deep learning. Authors [6] [10] usually work with recurrent neural networks (RNN, LSTM) and convolutional neural networks (CNN). The concept of the Transformer took shape in 2017. The Transformer is presented in the paper [24] which describes its architecture and performance on several translation datasets. To the best of our knowledge, its use remains exclusive to the field of natural language preprocessing. This motivated us to test and to compare the performance of the transformers against other statistical, machine learning and deep learning models in forecasting time series product data.

## 3 PROPOSED APPROACH

The proposed architecture, depicted in figure 1, aims at creating an intelligent model for product quality prediction for industry 4.0. Our chosen KPI is improving quality as it indicates the percentage of correctly manufactured products. It is a four-layer architecture, namely : data exploration layer where we prepare our data to be suitable for the model. A Second layer, representing the feature engineering step where we propose a feature selection and data balancing approach. A Third layer that consists of a comparison of different categories of techniques/algorithms for time series forecasting: statistics-based, machine learning-based and deep learning-based. And finally a forth layer dedicated to model interpretation helps adjust the production values.

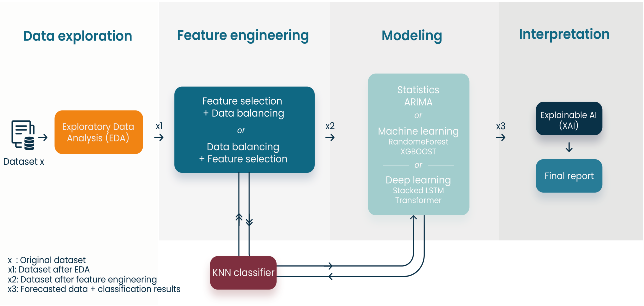

Data exploration
Feature engineering
Modeling
Interpretation
Dataset x
Exploratory Data Analysis (EDA)
x1
Feature selection + Data balancing
or
Data balancing + Feature selection
x2
Statistics ARIMA
or
Machine learning RandomeForest XGBOOST
or
Deep learning Stacked LSTM Transformer
x3
Explainable AI (XAI)
Final report
KNN classifier
x : Original dataset
x1: Dataset after EDA
x2: Dataset after feature engineering
x3: Forecasted data + classification results

**Figure 1: Proposed architecture**

In the following subs-sections the different layers of the architecture are detailed.

### 3.1 Data exploration

Data exploration refers to the first step of data analysis in which statistical techniques and visualisation are used to describe the characteristics of a dataset, such as size, quantity, distribution in order to better understand its nature. In our work, we followed the next steps. First, we proceeded with a raw data visualization. This showed us the distribution of the columns and their correlation. Second, we verified if the dataset contains missing, negative or null data. Third, we used quantile outlier removal to eliminate undesirable noise. We also counted the occurrences in the output class. Furthermore, since our dataset is a time series, the exploratory analysis expands to discover other properties of the data. A time series dataset is stationary when its statistical properties (expectation, variance, auto-correlation) are not time-varying. Having a stationary dataset means that it's free of trends and seasonality. In order to verify the stationarity of our time series, we implemented the Dickey-Fuller test [13].

### 3.2 Feature Engineering

In order to choose the most appropriate approach for our study case, we carried out a comparative analysis between the two following approaches: applying feature selection followed by data balancing or applying data balancing followed by feature selection. The approach with the best classification results is chosen. For this purpose, we implemented a KNN classifier.

*3.2.1 Feature selection.* The feature selection step consists of reducing the number of input variables when developing a predictive model. It is a basic technique for directing the use of variables to the most effective and efficient for a particular machine learning system. This practice allows the algorithm to adjust and learn more quickly and, more importantly, reduces its complexity in order to make it easier to interpret. According to [14], there are three feature selection techniques: filter method [15], wrapper method [16] and embedded method[17]. In our approach, we experimented all three techniques and compared their results to derive the best approach. We used as a filter method: variance threshold, as a wrapper method: XGBOOST with recursive feature elimination (rfe) and finally as an embedded method: lasso.

*3.2.2 Data balancing.* Among the techniques used in data balancing are: Oversampling which creates artificial instances of minority classes and undersampling which is used to eliminate the instances corresponding to the majority class. To balance the data, we opted for a hybrid approach which consists of applying oversampling to balance the data, followed by undersampling to remove any unwanted noise. The oversampling is performed by creating synthetic minority class samples to balance the dataset with SMOTE [18]. For the undersampling, we compared two techniques. The first one is tomekLinks (TL) [20] and the second technique is Edited Nearest Neighbours (ENN) [19].

### 3.3 Modeling

Time series forecasting is about making forecasts based on time-stamped historical data. It involves building models through historical analysis and using them to make observations and future policy decisions. We have chosen to compare three categories of algorithms and choose the most appropriate for our case. 1) Statistical model: ARIMA. 2) Machine learning models: Random forest and XGBOOST. 3) Deep learning models: Stacked LSTM and Transformer-based model.

*3.3.1 Statistical model: ARIMA.* The ARIMA model [21] is a statistical method used in stationary time series analysis and forecasting. ARIMA is an abbreviation for Auto Regressive Integrated Moving Average. It is a forecasting algorithm based on the idea that the information in the past values of the time series can alone be used to predict the future values.

*3.3.2 Machine learning models.* Machine learning techniques can be used for time series prediction. It is necessary to convert the time series prediction problem into a supervised learning problem. To create the new data form, the previous time steps are considered as input variables and the next time step is considered as output variable [22]. For our work, we chose to apply the random forest and XGBOOST algorithms for the prediction.

*3.3.3 Deep learning models.* **Stacked LSTM:** LSTM networks are a special type of RNN capable of learning from long-term dependencies. Stacking LSTM layers deepens the model as the upper LSTM layer provides the sequence output to the lower LSTM layer instead of the single value output. We propose the following model: LSTM(n=100)(x3), dense(n=50), LSTM(n=50)(x2) and a final dense(n=1). We used early stopping with patience equals 25. For the hyperparameters tuning, randomSearch was parameterized in order to find the combination that returns the minimal MAPE error. **Hyperparameters:** Learning_rate = 0.001, Batch_size = 128, Epochs = 61.

**Transformer-based model:** The Transformer [24] is a new network architecture based on a "Self-attention" mechanism. It returns data we need by focusing only on significant features of a single sequence. We propose a model inspired by the Transformer. The model consists of Transformer encoder blocks (x8) that use the MultiHeadAttention layer (x8) as a self-attention mechanism applied to the input data. The Transformer encoder block generates a batch_shape + (num_steps, features) tensor. This tensor is processed through a neural network (multilayer perceptron) and lastly through a dense layer with a relu activation function to produce the final output. We used early stopping with patience equals 25. RandomSearch was parameterized to in order to find the combination that returns the minimal MAPE error. **Hyperparameters:** batch_size= 64, dropout= 0.3, epochs= 300, head_size= 128, hidden_dim= 100, learning_rate= 0.005. For the multilayer perceptron: mlp_dropout= 0.2, mlp_units=64.

## 3.4 Interpretation

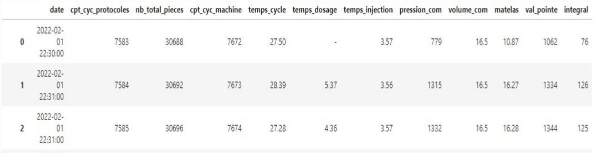
Machine learning models remain largely black boxes. However, understanding the reasoning behind the predictions is very important especially in a sensitive decision for industrial case. AI interpretability shows what is going on in these systems and helps identify potential problems and model errors. In our work, we used LIME [25] interpreter. It is based on finding independent descriptions for each instance by creating random samples in the area at regular intervals and weights according to the distance from the point of origin. The local description identifies which dimension of the input is most responsible for the output of the neural network. The algorithm provides a linear explanatory model and can be plotted for visualisation.

# 4 REAL CASE STUDY: EXPERIMENTATION AND RESULTS

The experiment is based on a real case. ADDIXO has provided us with quality related data of plastic product manufacturing. In this work, a product’s quality prediction module is to be integrated to the ADDIXO Smart Factory solution. We used the libraries Keras and scikit-learn with a TensorFlow backend along side with pandas for tabular data processing. All visuals are produced with matplotlib and seaborn.

## 4.1 Data exploration

The dataset, illustrated in figure 2, contains 16 columns and 94528 rows. The columns (1-15) represent process variables such as injection time, pressure and volume. The output column represents the product’s quality evaluation of the manufactured product where class 1 = OK and class 0 = Not OK. We proceeded with the following analysis: first the data visualization, described in figure 3, showed normally distributed variables as the majority of data points are relatively similar. This helped us identify the presence of outliers. We used quantile method to remove all unwanted noise. Second, the experimental dataset doesn’t contain neither missing nor null or negative values. Third, a count of the output class showed an unbalance. Finally according to the Dickey–Fuller test results, the dataset is stationary at 95% level of confidence.



| | date | cpt_cyc_protocoles | nb_total_pieces | cpt_cyc_machine | temps_cycle | temps_dosage | temps_injection | pression_com | volume_com | matelas | val_pointe | integral |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2022-02-01 22:30:00 | 7583 | 30688 | 7672 | 27.50 | - | 3.57 | 779 | 16.5 | 10.87 | 1062 | 76 |
| 1 | 2022-02-01 22:31:00 | 7584 | 30692 | 7673 | 28.39 | 5.37 | 3.56 | 1315 | 16.5 | 16.27 | 1334 | 126 |
| 2 | 2022-02-01 22:31:00 | 7585 | 30696 | 7674 | 27.28 | 4.36 | 3.57 | 1332 | 16.5 | 16.28 | 1344 | 125 |

**Figure 2: Sample data**

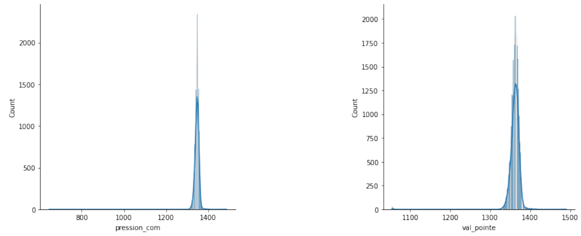


**Figure 3: Normal distribution**

## 4.2 Feature engineering

*4.2.1 Feature selection.* In order to choose the optimal number of features for the experimental dataset, we proceeded with the comparison of the 3 methods presented in **3.2.1**. New datasets

**Table 1: Selected features with three methods**

| Method | Selected features | Accuracy | ROC_AUC score |
|---|---|---|---|
| Original dataset | 16 | 0.9985 | 0.9827 |
| Filter | 8 | 0.9984 | 0.9636 |
| Wrapper | 15 | 0.9983 | 0.9504 |
| Embedded | 12 | **0.9994** | **0.9955** |

are generated with the given number of selected features. A KNN classifier is used to compare the new datasets along side with the original one. For the Classification task we evaluated our models with accuracy score and roc_auc score. We observe that in Table 1, the embedded method has scored the highest accuracy and roc_auc score.

4.2.2 *Data balancing.* A count of the target columns showed an unbalance in our dataset. Minority class (319 instances) is almost 133 time smaller than the majority class (42426 instances). We used imblearn.combine packages SMOTEENN and SMOTETomek. As their names show, SMOTEENN packages applies ENN undersampling after SMOTE oversampling. Same for SMOTETomek as it applies applies tomekLink undersampling after SMOTE oversampling. To compare the balancing results, we used a KNN classifier with the following parameters: metric= 'manhattan', k=3 and weights='distance'. The results are summarized in Table 2. Smote + tomekLinks has achieved the highest roc_auc score (0.92) whereas Smote + ENN has achieved the highest accuracy (0.95). Based on [27], roc_auc score is a better measure than accuracy. And so, we proceed with the generated new dataset. **Data balancing results:** Class 0: 26102 instances. Class 1: 55565 instances.
At this stage, after concluding the best techniques in feature selection (embedded method) as well as in data balancing (Smote+tomekLinks), we need to choose the best chronological order: (1) feature selection then data balancing or (2) data balancing then feature selection .

**Table 2: Data balancing results**

| Method | Accuracy | ROC_AUC score |
|---|---|---|
| SMOTE + ENN | **0.9566** | 0.8928 |
| SMOTE + tomekLinks | 0.9132 | **0.9293** |

**Table 3: (1) and (2) comparison results**

| Approach | Selected features | Accuracy | ROC_AUC score |
|---|---|---|---|
| (1) | 10 | **0.9994** | **0.9998** |
| (2) | 12 | 0.9956 | 0.8622 |

Table 3 shows that the approach (1) scores the best results as it returns 10 features out of 16 with accuracy and roc_auc score very close to 1. We conclude that, for our case study, best approach is feature selection followed by data balancing.

## 4.3 Modeling

Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean absolute percentage error (MAPE) are calculated to evaluate the models' forecasting results. Analysing table 4, we can conclude that the transformer-based model has achieved the best scores (2/3). This is because of its the ability to capture long-range dependencies and interactions. Plus since its design can allow parallel training of the data, the transformer-based model is more efficient than LSTM especially time wise. For our case, it is 11.32 times faster than the stacked LSTM. Random forest is a strong contender as it scored overall good values in a the minimal time of 2.89 seconds.

**Table 4: Forecasting results**

| Model | RMSE | MAPE | MAE | Time (s) |
|---|---|---|---|---|
| ARIMA | 2.864 | 0.067 | 2.043 | 2245.74 |
| XGBOOST | 1.689 | 0.025 | 0.791 | 5.335 |
| RandomForest | 1.615 | **0.023** | 0.724 | **2.891** |
| Stacked LSTM | **1.464** | 0.027 | 0.838 | 1143.38 |
| Transformer-based | 1.548 | **0.023** | **0.719** | 101 |

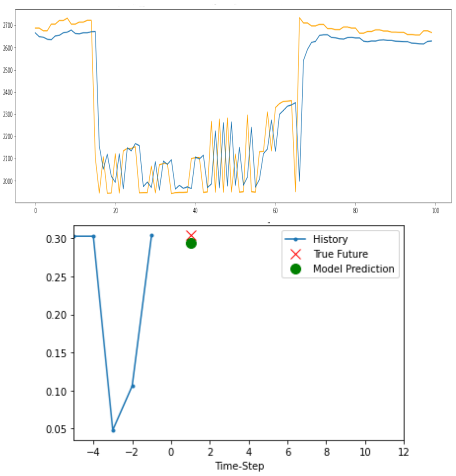


**Figure 4: Transformer-based model's predictions**

That's why we chose the transformer-based model to generate a new dataset of predicted values. Figure 4 represents our best model's prediction where the x axis represents the time steps and the y axis represents the values of the predicted column. Next we proceeded with the classification of this new data in order to check the quality of the product. We used a KNN classifier with the following parameters: metric= 'manhattan', k=3 and weights='distance'. **Classification results:** Dataset size: 12242, Class 0: 3558, Class 1: 8684

## 4.4 Interpretation

LIME returns a group of weights explaining how the model classified the random sample [25]. When the weight is positive, the designated variable favors the classification and vice versa. Comprehending how the model classified the sample helps us monitor the variables. Table 5 represents the recommended values to obtain class 1 that represents acceptable quality of the product.

**Table 5: Recommended values**

| Variables | Recommended values |
|---|---|
| Cycle time | 26.90 < x ≤ 26.94 |
| Dosing time | 4.01 < x ≤ 4.03 |
| Injection time | x ≤ 3.48 |
| Switching pressure | 1347.00 < x ≤ 1353.00 |
| Switching volume | x ≤ 16.50 |
| Mattress | 16.32 < x ≤ 16.35 |
| Peak value | 1363.00 < x ≤ 1369.00 |
| Integral value | x > 142.00 |
| Total pressure | 2710.00 < x ≤ 2721.00 |
| Total volume | 32.82 < x ≤ 32.85 |

To validate the obtained recommended values, two sets of data samples of the same size are fed into the architecture. The values of the first set are randomly collected, while the values of the second set were filtered to be in the range of the recommended values. The production of badly manufactured products (class 0) has decreased from 91.88% to 57.55% in the second set.

## 5 CONCLUSIONS

In this paper, we proposed an approach based on different AI models for product quality monitoring in an industrial context. Our chosen KPI is the percentage of correctly manufactured products. The data preprocessing phase showed a big unbalance between output classes. This inspired us to experiment two approaches for feature the engineering phase. The results have shown that feature selection followed by data balancing returns the best scores accuracy and roc_auc wise. Next, we proceeded with time series forecasting. The originality of our work consists in using a transformer-based model within a time series prediction problem for industry 4.0. The evaluation of all the phases using the appropriate methods and metrics has shown good results, in terms of accuracy, MAPE, RMSE and the over all execution time, which proves the effectiveness of our proposal. The transformer-based model scored the best (rmse= 1.305, mape=0.23, mae=0.548) along side with random forest (rmse= 1.615, mape=0.23, mae=0.724) that appeared to be a serious contender to deep learning techniques for forecasting. LIME interpreter was used to provide the recommended values. Our work is integrated as a module within ADDIXO's smart factory system in order to make early decisions about the product's quality. In our future work, we intend to experiment more industrial datasets with different variables in order to monitor other quality aspects. We also intend to use Temporal fusion transformer architecture as it has shown significant performance improvements over existing benchmarks.

## REFERENCES

[1] *Productivity and performance improvement from industry 4.0 adoption Indonesia* Statista Research Department: (2019)

[2] Jacob, D *Quality 4.0 impact and strategy handbook* LNS Research, MaterControl (2017)

[3] Chunquan Li, Yaqiong Chen, Yuling Shan *A review of industrial big data for decision making in intelligent manufacturing* Engineering Science and Technology, an International Journal, (2021)

[4] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone *Machine learning for predictive maintenance: A multiple classifier approach* IEEE Transactions on Industrial Informatics (2015)

[5] Changqing Liu, Yingguang Li, Guanyan Zhou, Weiming Shen *A sensor fusion and support vector machine based approach for recognition of complex machining conditions* Journal of Intelligent Manufacturing, Springer (2018)

[6] Yang Guo, Zhenyu Wu, Yang Ji *A hybrid deep representation learning model for time series classification and prediction* 3rd International Conference on Big Data Computing and Communications (BIGCOM) (2017)

[7] David Gyulai, Andras Pfeiffer, Gabor Nick, Viola Gallina *A hybrid deep representation learning model for time series classification and prediction* 3rd International Conference on Big Data Computing and Communications (BIGCOM) (2017)

[8] Jianjing Zhang, Peng Wang, Ruqiang Yan, Robert X. Gao *Long short-term memory for machine remaining life prediction* Journal of Manufacturing Systems (2017)

[9] Johannes Futterer, Maksymilia nKochanski, Dirk Muller *Application of selected supervised learning methods for time series classification in Building Automation and Control Systems* International ConferenceFuture Buildings Districts – Energy Efficiency from Nano to Urban Scale (2017)

[10] Chen, L, Xu, G, Zhang, S, Yan, W, Wu, Q *Health indicator construction of machinery based on end-to-end trainable convolution recurrent neural networks* Journal of Manufacturing Systems (2020)

[11] Stu Johnson *Quality 4.0: a trend within a trend* Quality magazine (2019)

[12] Andrew V. Metcalfe, Paul S.P. Cowpertwait *Introductory Time Series with R* Springer (2009)

[13] Giuseppe Schlitzer *Testing the stationarity of economic time series: further Monte Carlo evidence* Ricerche Economiche (1995)

[14] Ismael Ramos-Pérez, Álvar Arnaiz-González, Juan J. Rodríguez, César García-Osorio *When is resampling beneficial for feature selection with imbalanced wide data* Expert Systems with Applications (2021)

[15] Bommert, A, Sun, X, Bischl, B, Rahnenführer, J, Lang, M *Benchmark for filter methods for feature selection in high-dimensional classification data. Computational Statistics Data Analysis* Computational Statistics Data Analysis (2020)

[16] Kohavi, R., John, G. H. *Wrappers for feature subset selection. Artificial Intelligence* Artificial Intelligence (1997)

[17] Tarfa Hamed, Rozita Dara, Stefan C. Kremer *An accurate, fast embedded feature selection for SVMs* 13th International Conference on Machine Learning and Applications (2014)

[18] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP *SMOTE: synthetic minority oversampling technique* Journal Of Artificial Intelligence Research, Volume 16, (2002=

[19] Ferri, J.V. Albert, E. Vidal *Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules* IEEE Trans Syst Man Cybern B Cybern (1997)

[20] I. Tomek *Two modifications of CNN* IEEE Transactions on Systems, Man, and Cybernetics (1976)

[21] Zohair Malki, El-Sayed Atlam, Ashraf Ewis *ARIMA models for predicting the end of COVID-19 pandemic and the risk of second rebound* Neural Computing and Applications (2020)

[22] *How to Convert a Time Series to a Supervised Learning Problem in Python* machinelearningmastery.com

[23] Grace W. Lindsay *Attention in Psychology, Neuroscience, and Machine Learning* . Front. Comput. Neurosci (2020)

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin *Attention Is All You Need* NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems (2017)

[25] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin *Why Should I Trust You?": Explaining the Predictions of Any Classifier* KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)

[26] James Bergstra, Yoshua Bengio *Random Search for Hyper-Parameter Optimization* Journal of Machine Learning Research (2012)

[27] Charles X. Ling, Jin Huang, Harry Zhang *AUC: A Better Measure than Accuracy in Comparing Learning Algorithms* Conference of the Canadian Society for Computational Studies of Intelligence (2003)

# Grayscale Text Watermarking

Simone Branchetti
Department of CSE
University of Bologna
Bologna, Italy
simone.branchetti2@studio.unibo.it

Flavio Bertini
Department of MPCS,
University of Parma
Parma, Italy
flavio.bertini@unipr.it

Danilo Montesi
Department of CSE
University of Bologna
Bologna, Italy
danilo.montesi@unibo.it

## ABSTRACT

In an increasingly connected world, where information is easily spread through multiple channels and platforms, digital watermarking has been broadly investigated in authorship attribution and intellectual property protection of digital content. However, text contents pose many challenges due to a low capacity to embed a watermark. In this paper, we propose a new structural watermarking method for small pieces of text that may allow to hide upwards of 15 bits of watermark per character manipulating the underlying font grayscale values. The proposed method ensures length preservation and is robust to the copy and paste activities. Moreover, the method is able to embed a password-based watermark returning visually indistinguishable watermarked text.

## CCS CONCEPTS

• **Human-centered computing** → *Social content sharing*; Collaborative content creation; • **Information systems** → **Data provenance**;

## KEYWORDS

Grayscale Watermarking, Structural Text Watermarking, Digital Watermarking, Copyright Protection

**ACM Reference format:**
Simone Branchetti, Flavio Bertini, and Danilo Montesi. 2022. Grayscale Text Watermarking. In *Proceedings of International Database Engineered Applications Symposium, Budapest, Hungary, August 22–24, 2022 (IDEAS'22),* 5 pages.
https://doi.org/10.1145/3548785.3548794

## 1 INTRODUCTION

In the last decades, we have witnessed a rapid spread of numerous cloud platforms that meet several kinds of users' needs increasing the users' sharing behaviour of images, videos and text documents. The great availability of millions of digital content has unlocked research and business activities in heterogeneous domains using methodologies that require high data availability, such as data mining [5] and information retrieval [6]. On the flip side, the increased circulation of data and information has exacerbated issues related to data privacy and provenance also given the intellectual property rights that usually cover the digital content.

A common approach for digital content protection implies the use of watermarking techniques [17]. In particular, watermarking a piece of media means embedding information into it with the explicit intent of preserving the copyright and eventually tracking the origin. Out of all digital content, watermarking a piece of text instead of, for example, an image or video rises many more challenges. In particular, the text has low embedding bandwidth, meaning that there's much less room to embed the payload compared to an image, where every pixel can hide many bits of the watermark. Moreover, the text allows a restricted number of alternative syntactic and semantic permutations to preserve readability and original meaning [15]. In particular, text watermarking approaches can be classified into *zero watermarking techniques*, if some features of the text are stored on a third-party authority server; *image-based techniques*, if the text is transformed into an image and the watermark is embedded using an image watermarking method, for this reason, it cannot be considered a pure text watermarking method; *syntactic and semantic techniques*, that exploit the language depending features and grammar rules to embed the watermark; and *structural techniques*, that exploit structural and language-independent characteristics to embed the watermark.

In this paper, we present a new pure text watermarking method for small pieces of text which nevertheless works adequately with long texts also. The proposed structural method is able to embed a password-based watermark preserving the length of the original text and returning visually indistinguishable watermarked text. The method consists of three phases, that is the generation, embedding and resolution. The watermark is generated by applying a hash function that combines the text and the user's password. The embedding phase exploits the underlying font grayscale values, allowing to hide upwards of 15 bits of watermark per character. This represents our main contribution and improves the payload capabilities of state-of-the-art structural methods. In practice, the proposed method is to slightly change the black colour of every single character in the text using a shade of grey that is indistinguishable to the user and devote the bits freed up by this change to embedding the watermark bits. To evaluate the threshold of grey to be used, we held a specific survey showing how watermarked text is visually indistinguishable to the majority of people. The resolution phase extracts the watermark from the text allowing some collateral actions, such as the text integrity verification and the provenance of the textual data, since the extracted watermark is inextricably traceable back to the author thanks to its password. The proposed method has many significant features. It ensures length preservation, meaning that it does not cause overhead to the original document. The grayscale threshold ensures that the

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548794

resulting watermarked text is visually indistinguishable compared to the original one. Finally, it significantly increases the payload capability, which usually represents a weakness of text watermarking methods. Moreover, the proposed method can be programmatically applied to document editing software, such as Google Documents, Microsoft Word and LibreOffice Writer.

The rest of the paper is structured as follows. In Section 2, we discuss previous works related to text watermarking methods. In Section 3, we present our text watermarking method based on font grayscale values, whereas the held survey is described in Section 4. Results and limits are discussed in Section 5. Some concluding remarks are made in Section 6.

## 2 RELATED WORKS

In this section, we describe previous work in text watermarking, outlining drawbacks and limitations and excluding the zero-watermarking approach since it does not include a real watermark embedding phase.

The *image-based text watermarking* transforms the text into an image and embeds the watermark by modulating the pixels' luminance [4], the images histogram [11], or altering the inter-word spaces [10] and the characters' strokes and serifs [1]. Image-based methods reduce the text watermarking problem to the more researched scenario of image watermarking, actually making it unnatural and impractical in several contexts.

In *syntactic and semantic text watermarking*, the Natural Language Processing techniques exploit the syntactic and semantic structure of the text to embed the watermark. In particular, these approaches apply clefting/passivization [2] or morpho-syntactic transformations [12] or exploit the terms similarity [20] and nouns and verbs [18] to embed the watermark according to the sequence of bits. These methods make extensive modifications to the original text, producing a visibly different document and altering the author's content. Moreover, they require long text to embed the whole watermark.

*Structural text watermarking* is the most recent approach that exploits the underlying structure of the text to embed the watermark. In particular, given the increased diffusion of the Unicode standard in information systems, data hiding through Unicode transformation is recently receiving more interest in exploiting different whitespaces encoding [14], invisible symbols [13] and homoglyph characters [16]. Unlike the syntactic and semantic methods, the structural approach preserves the text content. However, a small set of homoglyphs can be effectively used, and the text font impacts the visual indistinguishability. For this reason, we proposed a structural method grayscale-based able to ensure visual indistinguishability and length preservation of the original text.

To the best of our knowledge, this is the first study that uses a font grayscale hue as a structural characteristic for text watermarking without transforming the text into an image as done in [4]. In the literature, the papers regarding grayscale have been focused mainly on grayscale recognition done by way of computer vision, focused on images [3]. Whenever grayscales are used for watermarking purposes, they are used to watermark images like in [8], or image-based approaches for text watermarking like in [7] where texts and images are watermarked together.

## 3 OUR METHOD

In this section, we present our structural Grayscale Text Watermarking (GTW) method, which works by embedding watermark bits into the shades of grey of the characters in a text document. Grayscale is a palette of colours that starts with pure black and ends with pure white, usually represented as a series of hexadecimal codes (e.g., #171717 is a shade of grey very closely resembling pure black #000000). After the process is complete each character has a different shade of grey and, as such, stands for different parts of the original bit sequence of the watermark.

The main problem that needs to be solved is that not all shades of grey are indistinguishable when confronted with pure black text. To solve this issue, GTW uses a two-step process to ensure maximum bit embedding per character while maintaining the outcome invisible to the majority of people. The first step is thresholding the maximum possible shades of grey we use to embed a sequence of bits in a character. This reduces the number of bits we can embed in a single character but improves indistinguishability compared to using the whole palette of grey or the whole colour spectrum. We will provide more details about the threshold and users' perceptions in Section 4.

The second step is the separation of the hexadecimal components of the character's colour (i.e., *red*, *green*, *blue*). If we used the entire colour spectrum and a maximum threshold of #070707, there would be several colours represented with smaller numerically values that would be visually further away from black and more visible to the user (e.g., the numerical value #00ffff representing the colour yellow is smaller than #070707). The solution is to consider a separate threshold of #07 for each colour component and then reconstruct the full colour to use for the character reassembling the components.

As shown in Figure 1, the shade of each component encodes a portion of the sequence of bits of the watermark and helps to form the grey shade respecting the defined threshold. In particular, the grayscale is chosen converting a part of the watermark bits into a hexadecimal number lower than the current colour component's threshold. Considering the previous #070707 threshold, the maximum number of embeddable bits into a single colour component is three. The binary value 111 is equal to 07 in hexadecimal.

The whole watermarking process consists of three phases. Firstly, in the watermark generation phase, a hash function combines the original text and the user's password to generate the watermark bit sequence. Depending on the used hashing function, the watermark will have a different length. Then, the watermark is embedded in the original text characters by modulating the grey value according to the bit sequence of the watermark, as shown in Figure 1. For

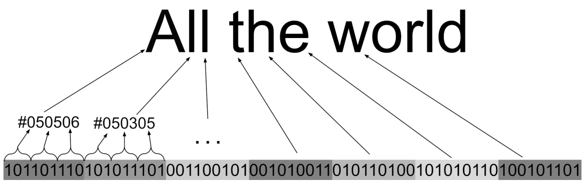


**Figure 1: Embedding of a 63bits long watermark in the first seven characters of the string.**

**Algorithm 1** Embedding

```
// The SHA256 hashing algorithm returns a 256bits watermark.
function EMBEDDING($orgText, userPassword$)
    $watermark \leftarrow SHA256(orgText, userPassword)$
    $threshold \leftarrow \#272727$
    for each $char \in orgText$ do
        $redBits \leftarrow toGreyShade(pop(watermark, 3))$
        $greenBits \leftarrow toGreyShade(pop(watermark, 3))$
        $blueBits \leftarrow toGreyShade(pop(watermark, 3))$
        $colour \leftarrow redBits + greenBits + blueBits$
        $wtmText \leftarrow wtmText + colouring(char, colour)$
    return $wtmText$
```

**Algorithm 2** Validation

```
// This function wipes out the watermark.
function CLEAN($wtmText$)
    for each $char \in wtmText$ do
        $text \leftarrow text + colouring(character, \#000000)$
    return text

// The extracted watermark is compared with the new one.
function VALIDATION($wtmText, newPassword$)
    $orgText \leftarrow clean(wtmText)$
    $newWtm \leftarrow SHA256(orgText, newPassword)$
    for each $char \in wtmText$ do
        $colourString \leftarrow string(getColor(char))$
        $red, green, blue \leftarrow extractComponent(colourString)$
        $recoveredWtm \leftarrow recoveredWtm + red + green + blue$
    return $(newWtm == recoveredWtm)\ ?\ True : False$
```

instance, using #070707 as a threshold value, that is #07 for each colour component (i.e., *red*, *green*, *blue*), we can encode 3 bits of the watermark in each component embedding 9 bits on each character. Moreover, by embedding the watermark using grayscale manipulation, no overhead data is added to the original text, thus ensuring length preservation. The third phase is the watermark validation, which goal is to verify whether a user who claims ownership of the document is the actual author. The proposed method belongs to the blind text watermarking method class [15], which means that the watermark can be extracted without the original text. Because the original text is supposed to use black colour characters, while our watermarked text contains also characters with shades of grey, the watermark can be extracted from the watermarked text using a reverse colouring function. In practice, the extracted watermark is compared with the watermark generated by combining the original text, which is obtained by cleaning the watermarked text, and the password of the claiming user. Another consequence of the proposed approach is that the watermark is invisible, not readable and detectable [15]. Hence, it is sufficient to examine the colour of each character to determine whether it is pure black or a shade of grey.

The pseudo-code of the embedding and validation algorithms are presented in Algorithms 1 and 2, respectively. We used the SHA256 (Secure Hash Algorithms) hashing function that returns a 256bits long watermark. If the user has less stringent security needs, this function can be replaced with algorithms that generate shorter sequences. Whereas the #272727 threshold allows embedding 5 bits in each colour component, this means that the SHA256 digest can be embedded in 18 characters. Both the embedding and validation algorithms work at the character level, embedding and recovering the bits sequence of the watermark in the colour components of the characters. The cost of the two algorithms is linear on the number of characters in the text in the worst-case scenario. It is worth noticing that the proposed method is both languages independent and not bound to any platform. In other words, the method can be integrated into any text editing software that permits programmatically manipulating a single character's colour freely.

## 4 GRAYSCALE PERCEPTION SURVEY

In this section, we briefly present the survey through which we identified the grayscale threshold to be used to ensure the visual indistinguishability of the watermarked text.

We devised a survey to study how people perceive different shades of grey in text. In particular, we administered the survey to 255 people (114 females and 111 males) covering various age groups: 13.3% under 19 years old, 29.3% between 20 and 29, 7.6% between 30 and 39, 12.0% between 40 and 49, 30.2% between 50 and 59, 6.7% between 60 and 69 and 0.9% over 70 years old.

The survey is designed in two parts[1]. The first one consists of 12 questions with pairs of squares shown side-by-side, in which randomly one of the two squares is always pure black (#000000 ■) while the other changes from question to question at regular intervals to a lighter shade of grey from #070707 ■ to #5f5f5f ■. We then asked the participants whether they thought the two squares are both coloured black. The assumption is that if a user recognises one shade of grey as different from black, he/she will also identify all the lighter grey nuances. These preliminary results using squares made it possible to reduce the threshold search space. Figure 2 shows that more than 70% of people can correctly identify the difference between #272727 ■ and #000000 ■. In other words, the results suggest that we should aim for a lower maximum hue to minimise the watermark's visibility.

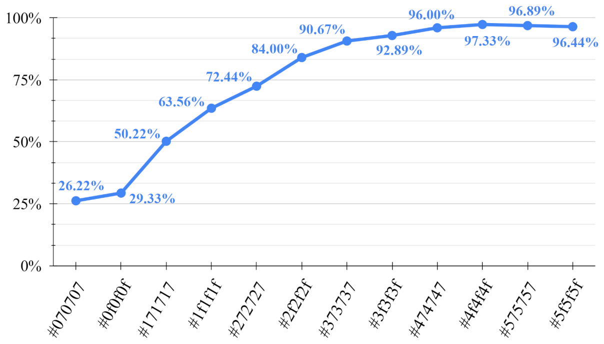
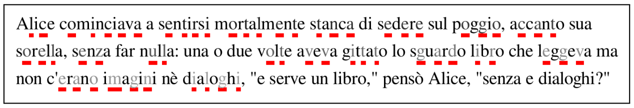


**Figure 2: The percentage of users who detect the two squares as different as the shade of grey changes.**

[1] The survey dataset is available at this link: https://tinyurl.com/22s6nvem.



Alice cominciava a sentirsi mortalmente stanca di sedere sul poggio, accanto sua sorella, senza far nulla: una o due volte aveva gittato lo sguardo libro che leggeva ma non c'erano imagini nè dialoghi, "e serve un libro," pensò Alice, "senza e dialoghi?"

**Figure 3: An example of text shown in the second part of the survey. The non-black letters have been underlined in red.**

The second part of the survey is designed to investigate the visibility of different shades of grey applied to written text. In particular, participants were asked to identify the first word in a series of six texts where they notice a different coloured letter, namely a letter with a different shade of grey. The texts are written in pure black and then some letters are recoloured in progressively lighter shades of grey, moreover, the texts vary in font size and boldness to mimic a good variety of real case scenarios. Figure 3 shows an example of text presented to participants. Each word in the text appears uniquely to be able to correctly identify the user's choice. To make it easier for the reader to identify letters with a different shade of grey, they have been underlined here in red.

Figure 4 shows the aggregated results of the second part. It is worth noticing that, across all font sizes, there is a sharp increase in recognition after the shade of grey #272727 ■, confirming that perception is different when shades of grey are applied to the text and allows higher threshold values. In particular, using a threshold value of #070707 ■ only 3.7% of the survey's population correctly identifies grayscale variations, the percentage rising to 9.08% with the threshold value of #272727 ■. This is an acceptable range of detection with thresholds that still allow us to hide a large number of bits in a short text.

## 5 RESULTS AND DISCUSSION

In this section, we discuss the crucial aspects concerning the proposed method, such as limits to preserve visual indistinguishability, performances in terms of payload capability and robustness of the watermark. In particular, we tested the implementation of the proposed method by writing a Google Documents add-on that automatically watermarks and verifies text documents of the users.

**Visual Indistinguishability at the Edges** - In some borderline cases of the sequences of bits, the visual indistinguishability of the watermark may be invalidated even with more restrictive threshold values. In particular, some unfortunate combinations of watermark bits can result in hexadecimal code values corresponding to more visible shades of the primary colours (i.e., *red*, *green*, *blue*) and not to shades of grey. For instance, the bits sequence 111110000000000 000001111100000 000000000011111 corresponds to three hexadecimal colour values to be assigned to three consecutive letters completely unbalanced among the colour components. In particular, the predominance of one of them in each chunk (e.g., the red component in the first of the three chunks) potentially results in a more noticeable text. Establishing the distance between colours is challenging and beyond the scope of this study, so we took a more conservative approach by reducing the number of bits that can be accommodated in each character. In particular, we imposed that the embeddable sequence of bits allowed by the threshold is repeated for all components. This ensures that every subsequence of bits of the watermark creates a shade of grey.

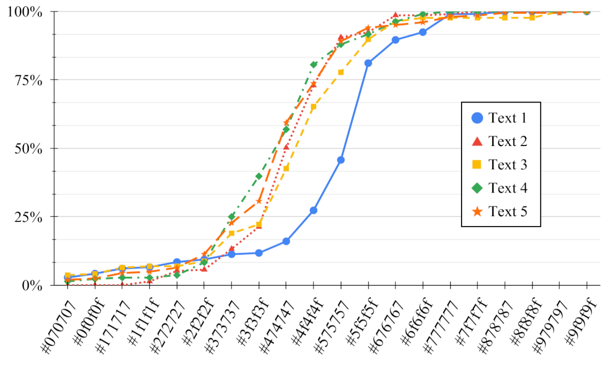


**Figure 4: Success identification rate vs shade of grey for each text from smallest (Text 1) to largest font size (Text 5).**

**Embedding Performance** - To evaluate the embedding performance with the state-of-the-art methods, we also selected a structural text watermarking method based on homoglyphs [15], which is a good benchmark in terms of payload and visual indistinguishability. The Homoglyph-based Watermarking (HBW) method embeds the watermark by swapping some characters and whitespace with characters that look the same but have a different Unicode representation. In particular, we used the New York Times Corpus[2] consisting of 1.8 million articles from the New York Times newspaper spanning from 1987 to 2007. We focused on the lead paragraph stressing the fact that the proposed method can be successfully applied to a short portion of texts. The GTW is evaluated using two different threshold values, that is #070707 and #272727. Whereas, since the hash functions ensure different levels of security, we used three different hash functions (e.g., SipHash, MD5 and SHA256) that provide different levels of security and generate sequences of bits of increasing length. Table 1 shows how many characters are needed on average to embed varying lengths of watermark bits sequence. The proposed GTW method with the most restrictive threshold value #070707 outperforms the HBW method. In particular, GTW is able to embed the longest sequence of bits (256bits) using 15 fewer characters than HBW when used to embed the shortest sequence of bits (64bits). Whereas the GTW method with the threshold value #272727 improves the embedding capabilities of the HBW method by 86.48% on average, showing the effectiveness of our approach. It is worth noticing that the GTW and HBW methods are not mutually exclusive and can be combined to achieve a higher embedding rate. As shown in Table 1, the combination requires just 46 characters to embed a 256bits long watermark, corresponding to 5 more characters than the table caption.

**Robustness** - In this section, we discuss the robustness of the proposed method against the most common attacks. Taking advantage of embedding capability and the ability to repeatedly embed the watermark in the text make the proposed method robust against insertion and deletion attacks. For instance, until there are 52 unaltered consecutive characters there will be at least one copy of the 256bits long watermark to retrieve in the text. The average word length in New York Times articles is 4.9 characters, this means that

[2] https://catalog.ldc.upenn.edu/LDC2008T19

**Table 1: Comparison results with state-of-the-art methods.**

| | **Required characters for:** | | |
|---|---|---|---|
| **Method** (*threshold*) | **64 bits** | **128 bits** | **256 bits** |
| GTW (*#070707*) | 22 | 43 | 86 |
| GTW (*#272727*) | 13 | 26 | 52 |
| HBW [15] | 101 | 198 | 357 |
| GTW (*#272727*) + HBW [15] | 12 | 23 | 46 |
| Khosravi et al. [9] | 2,133 | 4,267 | 8,533 |
| Por et al. [14] | 199 | 399 | 798 |
| Taleby A. et al. [19] | 1,016 | 2,032 | 4,064 |

the most robust watermark can be hidden in 18 words using the most stringent threshold, much less than the limit set by UK government best practice[3]. In other words, insertion and deletion attacks would require heavy changes to the watermarked text to effectively remove the mark leading to a completely different text with no longer any connection to the original text. The partial copy&paste attack is quite common and consists of an attacker copying and pasting part of the watermarked text violating the copyright of the author. Most structural watermarking techniques fail to protect against this type of attack. In [14], just 0.01% of the watermark is preserved against copy&paste. Whereas the GTW method performs excellently and, with 13 characters needed for embedding a 64bits long watermark, protects text at sentence/word level, making the partial copy&paste attacks ineffective. The text replacing attack can be considered a variant of the previous ones. In particular, the proposed approach outperforms other structural methods which shows a 2.6% success rate against this type of attack [9]. As with any other structural or image-based method, the GTW is vulnerable to manual retyping and letter recolouring attacks since producing new text by copying or recolouring characters completely wipes out the watermark. However, a partial recolouring attack can be considered equivalent to a replacement attack. It is worth noticing that the GTW method does not limit the user's ability to use different colours in the text document, for instance writing important words in red or blue. In particular, the solution is similar to the current method and involves the use of the shades of the colour used by the user to embed the watermark or, more simply, skip those words. Another important aspect concerning text watermarking is portability. In particular, we copied and pasted a watermarked piece of text to and from some of the major text editing software: Microsoft Word, LibreOffice Writer and Apache OpenOffice Writer. The results showed that all copy and paste attempts preserved the watermark since all of these software use the same hexadecimal scale to represent colour and share a protocol to support copy and paste action preserving the text formatting.

## 6 CONCLUSIONS

The provenance detection and intellectual property protection of digital content have become a challenging research problem, especially due to the increasingly widespread use of sharing platforms. In this paper, we proposed a structural text watermarking method that works by potentially embedding up to 15 bits into every single character changing the font colour from pure black into a grayscale hue chosen to be invisible to the human eye. We demonstrated that the shortest length to embed a 64-bit watermark is only 13 characters. The strengths of the proposed method include the preservation of the length of the original text, the guarantee that the change is visually indistinguishable, and the fact that the syntactic and semantic structure of the original text is unchanged. The proposed method, as well as being independent of the language used in the text and the user's preferred font, is fully portable to every major text editing software currently available.

[3] https://www.gov.uk/guidance/content-design/writing-for-gov-uk

## REFERENCES

[1] Tomio Amano and Daigo Misaki. 1999. A feature calibration method for watermarking of document images. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*. IEEE, 91–94.

[2] Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *International Workshop on Information Hiding*. Springer, 185–200.

[3] Jinfeng Bai, Zhineng Chen, Bailan Feng, and Bo Xu. 2014. Chinese image text recognition on grayscale pixels. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1380–1384.

[4] Anoop K Bhattacharjya and Hakan Ancin. 1999. Data embedding in text for a copier system. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, Vol. 2. IEEE, 245–249.

[5] Diogo Campos, Rodrigo Rocha Silva, and Jorge Bernardino. 2019. Text Mining in Hotel Reviews: Impact of Words Restriction in Text Classification.. In *KDIR*. 442–449.

[6] Alfredo Cuzzocrea, Wookey Lee, and Carson K Leung. 2015. High-recall information retrieval from linked big data. In *2015 IEEE 39th Annual Computer Software and Applications Conference*, Vol. 2. IEEE, 712–717.

[7] Anita John Jaseena K.U. 2011. Text Watermarking using Combined Image and Text for Authentication and Protection. *Int. Journal of Computer Applications* 20, 4 (2011). https://doi.org/10.1.1.206.4500

[8] Dr.Varghese Paul Jobin Abraham. 2011. Watermarking Grayscale Images using Text for Copyright Protection. *International Journal of Computer Applications* 31, 9 (2011). https://doi.org/10.1.1.735.1015

[9] Behrooz Khosravi, Behnam Khosravi, Bahman Khosravi, and Khashayar Nazarkardeh. 2019. A new method for pdf steganography in justified texts. *Journal of information security and applications* 45 (2019), 61–70.

[10] Young-Won Kim, Kyung-Ae Moon, and Il-Seok Oh. 2003. A Text Watermarking Algorithm based on Word Classification and Inter-word Space Statistics.. In *ICDAR*. Citeseer, 775–779.

[11] Young-Won Kim and Il-Seok Oh. 2004. Watermarking text document images using edge direction histograms. *Pattern Recognition Letters* 25, 11 (2004), 1243–1251.

[12] Hasan Mesut Meral, Bülent Sankur, A Sumru Özsoy, Tunga Güngör, and Emre Sevinç. 2009. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language* 23, 1 (2009), 107–125.

[13] Nighat Mir. 2014. Copyright for web content using invisible text watermarking. *Computers in Human Behavior* 30 (2014), 648–653.

[14] Lip Yee Por, KokSheik Wong, and Kok Onn Chee. 2012. UniSpaCh: A text-based data hiding method using Unicode space characters. *J. of Systems and Software* 85, 5 (2012), 1075–1082.

[15] Stefano Giovanni Rizzo, Flavio Bertini, and Danilo Montesi. 2019. Fine-grain watermarking for intellectual property protection. *EURASIP Journal on Information Security* 2019, 1 (2019), 1–20.

[16] Stefano Giovanni Rizzo, Flavio Bertini, Danilo Montesi, and Carlo Stomeo. 2017. Text watermarking in social media. In *Proc. of the 2017 IEEE/ACM Inter. Conf. on Advances in Social Networks Analysis and Mining 2017*. 208–211.

[17] Kartik U Sharma, Pooja P Talan, Pratiksha P Nawade, Mir Sadique Ali, and Akshay U Sharma. 2019. Digital Watermarking—An Overview and a Possible Solution. *ICTIS* (2019), 447–455.

[18] Xingming Sun and Alex Jessey Asiimwe. 2005. Noun-verb based technique of text watermarking using recursive decent semantic net parsers. In *International Conference on Natural Computation*. Springer, 968–971.

[19] Milad Taleby Ahvanooey, Hassan Dana Mazraeh, and Seyed Hashem Tabasi. 2016. An innovative technique for web text watermarking (AITW). *Information Security Journal: A Global Perspective* 25, 4-6 (2016), 191–196.

[20] Umut Topkara, Mercan Topkara, and Mikhail J Atallah. 2006. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th workshop on Multimedia and security*. 164–174.

# Measuring Vowel Harmony within Hungarian, the Indus Valley Script Language, Spanish and Turkish Using ERGM

Josey Vanorsdale
Department of Sociology, University of Nebraska-Lincoln
vanorsdale@huskers.unl.edu

Jigyasa Chauhan
School of Computing, University of Nebraska-Lincoln
jchauhan2@huskers.unl.edu

Sai Vivek Potlapally
School of Computing, University of Nebraska-Lincoln
spotlapally2@huskers.unl.edu

Srikar Chanamolu
School of Computing, University of Nebraska-Lincoln
schanamolu2@huskers.unl.edu

Sai Pratyush Reddy Kasara
School of Computing, University of Nebraska-Lincoln
skasara2@huskers.unl.edu

Peter Z. Revesz*
School of Computing, University of Nebraska-Lincoln
revesz@cse.unl.edu

## ABSTRACT

Front-back vowel harmony is an important characteristic of many languages. Testing whether an untranslated script has vowel harmony may aid its decipherment. This paper tests vowel harmony for three different modern languages (Hungarian, Spanish and Turkish) as well as the extinct underlying language of the undeciphered Indus Valley script. We also introduce a novel vowel harmony index based on the Exponential Random Graph Model for graphs. To achieve this, we first select words from each of the modern languages (Hungarian, Turkish, and Spanish) from their Swadesh list. Then we divide each word into syllables, isolating the vowels. We then analyze the three modern languages using Exponential Random Graph Model methods. The results indicate that this procedure and the vowel harmony index are feasible to define the degree of vowel harmony in a language. The procedure is then extended to the undeciphered Indus Valley Script. Our results indicate that the underlying language of the Indus Valley Script also had vowel harmony. We found that on average the odds of the IVS having vowel harmony were 6.61 times higher than would be found in a random graph.

## CCS CONCEPTS

• **Information systems** → Information systems applications; Data mining.

## KEYWORDS

Exponential random graph model, Front-back vowel harmony, Indus Valley script, Minoan, Odds ratio, Vowel harmony measure

**ACM Reference Format:**
Josey Vanorsdale, Jigyasa Chauhan, Sai Vivek Potlapally, Srikar Chanamolu, Sai Pratyush Reddy Kasara, and Peter Z. Revesz. 2022. Measuring Vowel Harmony within Hungarian, the Indus Valley Script Language, Spanish and Turkish Using ERGM. In *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3548785.3548788

## 1 INTRODUCTION

In some languages the vowels within words tend to be paired with each other if they are formed at the same area of the mouth. For example, in English the following vowels are formed at the back of the mouth: a, o, u, while the following vowels are formed at the front of the mouth: e, i. For example, *banana* has only back vowels, while *cherry* has only front vowels. A strong tendency towards front-back vowel harmony is common in Turkic and Uralic but is infrequent in Indo-European languages [1]. Front-back vowel harmony is assumed to have been a feature already in the Proto-Uralic language [4, 16] and can also be detected in the extinct Minoan language [9].

In this paper, we propose a new exponential random graph model (ERGM)-based measure for the degree of vowel harmony in languages and use that measure to compare three modern languages (Hungarian, Spanish, and Turkish) from three different language families (Uralic, Indo-European, and Turkic, respectively) and the underlying language of the Indus Valley Script, which is still considered undeciphered.

The rest of this paper is organized as follows. First, in the next section, we describe the data sources. Second, in the following section, we explain the use of simulated annealing for the Indus Valley Script. Third, we describe the exponential random graph model (ERGM) method [13] and proposes the odds ratio coefficient for 'nodematch' in an ERGM fit as a vowel harmony index. Fourth, we present and discuss the experimental results. Since the presence of front-back vowel harmony within the three modern languages (Hungarian, Spanish, and Turkish) has been already known, the focus is to answer the question of where the underlying language of the Indus Valley Script fits in. Fifth, in the last section, we give some conclusions and directions for future work.

## 2 DATA SOURCES

For Hungarian, Spanish and Turkish, we started with their Swadesh lists, which contained the 207 most basic words in those languages. Hungarian words were hyphenated by a native speaker, who also

*Corresponding author. Please send correspondence to revesz@cse.unl.edu.

*IDEAS'22, August 22–24, 2022, Budapest, Hungary*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9709-4/22/08…$15.00
https://doi.org/10.1145/3548785.3548788

Table 1: Statistics about the words considered and selected for the analysis.

| Language | Words Considered | Multisyllabic Root Words |
|---|---|---|
| Hungarian | Swadesh list (207 words) | 87 |
| Indus Valley Script | ICIT inscriptions | 61 |
| Spanish | Swadesh list (207 words) | 156 |
| Turkish | Swadesh list (207 words) | 82 |

identified and removed some suffixes. Spanish words were hyphenated by a tool called 'Hypenator.net' available at the Internet, and Turkish words were hyphenated using Wikipedia. The Spanish and the Turkish Swadesh list words were assumed to be root words. For the Indus Valley Script (IVS), we assumed that each sign represents a syllable. From the Interactive Corpus of Indus Text (ICIT) database of Indus Valley Script inscriptions of Well and Fuls [15], we selected those putative multisyllabic root words that appeared in at least four inscriptions. Table 1 shows the number of multisyllabic root words used in the analysis.

## 3 SIMULATED ANNEALING

For the Indus Valley Script, we use a simulated annealing process to determine the most likely front/back label for each node. Each node was randomly assigned a front or back designation. We, then, went through and randomly selected 200 nodes, and evaluated the node's neighbors. If it had more front vowel neighbors, then the symbol was changed to also be a front vowel. If the node had more back neighbors, it was also changed to a back vowel. Since there were 61 distinct symbols some nodes were evaluated more than once. After the 200 changes were made, a graph was made consisting of the probable front/back distributions. One example graph can be found in the upper right part of Figure 1.

## 4 EXPONENTIAL RANDOM GRAPH MODEL ANALYSIS

For each language, we create a graph where each node is a syllable, and each edge means that there is at least one root word which contains the syllables associated with the nodes that are connected. The nodes of the graph are labeled as belonging to front or back categories depending on whether the vowel in the syllable a front or a back vowel is.

The graph is analyzed using an exponential random graph model (ERGM) tool called STATNET and ERGM packages in R programming with a parameter for edges and 'nodematch' [13]. These calculate a coefficient that represents the degree to which the graph possesses more edges between nodes that have matching labels compared to a random graph. We call this coefficient the *vowel harmony index*.

## 5 EXPERIMENTAL RESULTS AND DISCUSSION

Our experiment had two main steps. First, we created a dataset for each language. Second, we created graphs for the words of each language and find the vowel harmony index. The results are summarized in Table 2.

**Table 2: The odds ratio, the log odds, and the significance value (P of .001 means 95 percent confidence) according to the ERGM analysis.**

| Language | Log odds | Odds ratio | Significance |
|---|---|---|---|
| Hungarian | 1.832 | 6.245 | P < .001 |
| Indus Valley Script | 1.889 | 6.61 | P < .001 |
| Spanish | -0.1562 | 0.855 | 0.231 |
| Turkish | 1.855 | 6.389 | P < .001 |

For Hungarian, the graph has one main component of front vowels and many smaller components of either only front vowels or only back vowels (upper left of Figure 1). Hence this graph indicates a strong vowel harmony.

For the underlying language of the Indus Valley Script, the graph has one large component and two small components with three and two nodes only. After simulated annealing tried to optimize the node labeling as back or front syllabic, the large component contained about the same number of front and back labeled nodes (upper right of Figure 1). Hence the underlying language of the IVS does not appear to have vowel harmony.

For Spanish, the graph has one large cluster with many front and back vowels (lower left of Figure 1). While some of the lone pairs appear to be matched on front/back vowels, there is clear evidence suggesting that Spanish does not have vowel harmony.

For Turkish, the graph has two main components (lower right of Figure 1). One component has only front vowels, while the other component has mostly back vowels. Hence this graph shows a strong degree of front-back vowel harmony.

In addition to the visual analysis, an ERGM analysis also calculated the log odds and odds ratios for the four languages. The odds ratio is the likelihood of each node having edges with a node that matched in the front/back label. This coefficient is our vowel harmony index. Its log is called the log odds, which is an alternative measure of vowel harmony.

For Hungarian, nodes were 6.245 times more likely to have edges to other nodes with the same front/back label, compared to a random graph. The same result can be interpreted as nodes having 83.2 percent higher odds of connecting to other nodes with the same front/back label. This was a significant result.

For the underlying language of the IVS, after 2000 iterations of simulated annealing performed repeatedly, the odds ratios ranged from a high of 26.90 to a low of 2.22 with an average of 6.61 of nodes matching in front/back label compared to a random graph. These results suggest that there is a high likelihood that the underlying language of the IVS has vowel harmony.

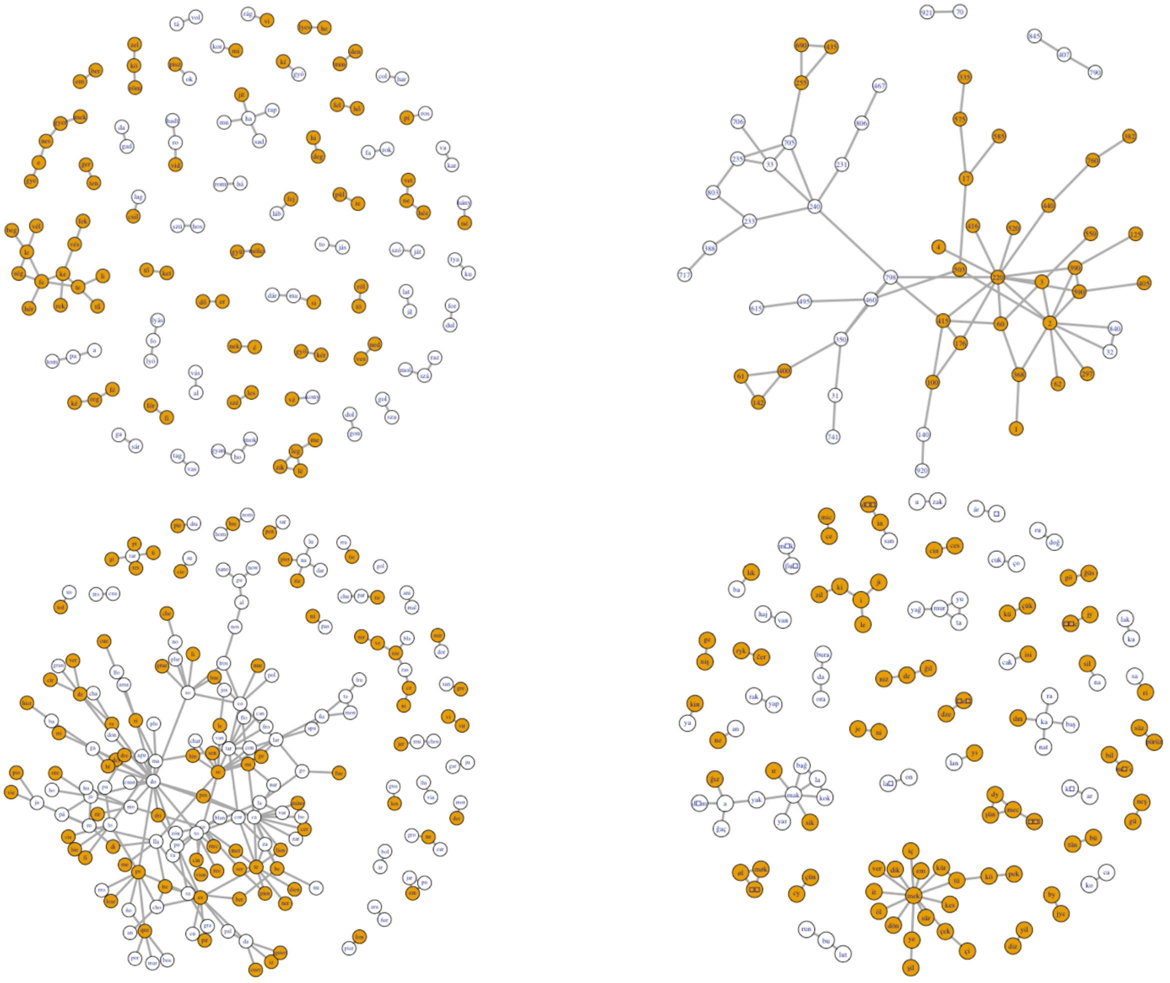


**Figure 1: ERGM analysis of Hungarian (upper left), the underlying language of the Indus Valley Script (upper right), Spanish (lower left), and Turkish (lower right). Syllables with back vowels (white), syllables with front vowels (orange).**

For Spanish, nodes had about 15.62 percent higher odds of connecting to other nodes of the opposite front/back label. These were not significant results with P = 0.231, suggesting that this relationship was about the same as would be found in a random graph.

For Turkish, nodes were 6.389 times more likely to have edges to other nodes with the same front/back label compared to a random graph. This was also a significant result.

Comparing the relative strength of vowel harmony in the four languages, we see that the underlying language of the Indus Valley Script has the highest vowel harmony index (6.61), Turkish is second (6.389), Hungarian is third (6.245). The difference in the vowel harmony index is small among these three languages. This indicates that the three languages all have strong front-back vowel harmony. Spanish had the smallest vowel harmony index (0.855), which is close to one, which is a value expected for random graphs. Hence the vowel harmony index suggests that Spanish does not have vowel harmony.

## 5.1 Comparison with the Minoan Language

The Minoan civilization was a contemporary of the Indus Valley civilization in the Bronze Age. Both civilizations left some inscriptions that attracted the attention of many scholars, who tried to decipher them. Decipherment of a script is facilitated by knowing the major characteristics of the underlying language. The presence or absence of front-back vowel harmony is one such characteristic. If we can identify several common characteristics between the Indus Valley and the Minoan languages, then the probability is high that the two languages were relatives, that is, belonged to the same language family.

The surprising result of this paper is that the underlying language of the Indus Valley Script seems to also have front-back vowel harmony. Hence the closest relative of the still undeciphered language of the Indus Valley Script is likely found among those languages that have front-back vowel harmony. Vowel harmony of the underlying language of the Minoan scripts was studied in Revesz [9], which showed that the Phaistos Disk had front-back vowel harmony. The

Phaistos Disk is the longest Minoan inscription with a total of 241 signs that are stamped left-to-right in spiral form [11]. That the Indus Valley and the Minoan languages both had front-back vowel harmony suggests that the two languages may have been related. There are other interesting connections between the Indus Valley and the Minoan civilizations. Tsafou and García-Granero [14] found that the Minoans used cumin *(Cuminum cyminum)* that originated from the Indus Valley civilization. Ialongo et al. [5] argued that Near Eastern traders were the middleman in the trade between the Indus Valley and the Minoan civilizations. However, Revesz, a data scientist, reanalyzed the weight unit data of Ialongo et al. [5] and showed that there was direct trade between the Indus Valley and the Minoan civilizations [12]. Revesz [12] also showed that there was a direct trade route between the Indus Valley and the Old European cultures, and the Minoan and Old European cultures too. This triangular connection raises the possibility that the Indus Valley, the Minoan, and the Old European cultures have some linguistic connections. Minoan and Hungarian belong to the same branch of the Uralic language family based on decipherment of some Minoan inscriptions [7] and regular sound changes [10], and their common ancestor may have been the language of the Old European culture. Similarities between Hungarian folk songs and Sanskrit literature also support the triangular connection among the three regions [8]. The similar motifs found in the Sanskrit literature may derive from the Indus Valley civilization [6].

Daggumati and Revesz [2] found the Indus Valley script to be closest to the Sumerian pictograms among a set of ancient scripts. However, an interesting similarity between the Indus Valley script and the Minoan Linear A script is that they both contain many allographs [3]. The direct trade between the two civilizations and this aspect of their scripts further increases the probability that these two civilizations spoke a similar language.

## 6 CONCLUSIONS AND FUTURE WORK

The ERGM odds ratio coefficient using 'nodematch' seems to be a suitable *vowel harmony index*. The vowel harmony index gives results that match expectations regarding several modern languages from three different language families.

The closest relative of the still undeciphered language of the Indus Valley Script is likely found among those languages that have front-back vowel harmony, and it may turn out to be the Minoan language. This recognition could direct future work on the decipherment of the Indus Valley Script together with other significant analyses of the structure of the Indus Valley Script signs and inscriptions.

## REFERENCES

[1] Haruo Aoki. 1968. Toward a topology of vowel harmony. International Journal of American Linguistics, 34 (2), 142-145.

[2] Shruti Daggumati and Peter Z. Revesz. 2018. Data Mining Ancient Script Image Data Using Convolutional Neural Networks. In Proceedings of the 22nd International Database Engineering amp; Applications Symposium (IDEAS 2018). ACM Press, New York, NY, USA, 267-272. https://doi.org/10.1145/3216122.3216163

[3] Shruti Daggumati and Peter Z. Revesz. 2021. A method of identifying allographs in undeciphered scripts and its application to the Indus Valley Script. Humanities and Social Sciences Communications, 8, 50. https://doi.org/10.1057/s41599-021-00713-0

[4] Robert Thomas Harms. 2021. Linguistic Characteristics. Encyclopedia Britannica. https://www.britannica.com/topic/Uralic-

[5] Nicola Ialongo, Raphael Hermann, and Lorenz Rahmstorf. 2021. Bronze Age weight systems as a measure of market integration in Western Eurasia, Proceedings of the National Academy of Sciences, 118, 27. https://doi.org/10.1073/pnas.2105873118

[6] Asko Parpola, 2015. The roots of Hinduism: The early Aryans and the Indus civilization. Oxford University Press, USA.

[7] Peter Z. Revesz, 2017. Establishing the West-Ugric language family with Minoan, Hattic and Hungarian by a decipherment of Linear A. WSEAS Transactions on Information Science and Applications, 14, 306-335.

[8] Peter Z. Revesz, 2019. A comparative analysis of Hungarian folk songs and Sanskrit literature using motif similarity matrices. WSEAS Transactions on Information Science and Applications, 16, 75-86.

[9] Peter Revesz. 2020. A Vowel Harmony Testing Algorithm to Aid in Ancient Script Decipherment. In Proceedings of the 24th International Conference on Circuits, Systems, Communications and Computers (CSCC'20), IEEE Press, Piscataway, NJ, 35-38.

[10] Peter Z. Revesz, 2020. Minoan and Finno-Ugric regular sound changes discovered by data mining. In Proceedings of the 24th International Conference on Circuits, Systems, Communications and Computers (CSCC'20), IEEE Press, Piscataway, NJ, 241-246.

[11] Peter Z. Revesz. 2022. Experimental evidence for a left-to-right reading direction of the Phaistos Disk. Mediterranean Archaeology and Archaeometry, 22 (1), 79-96. https://doi.org/10.5281/zenodo.6311386

[12] Peter Z. Revesz. 2022. Data science applied to discover ancient Minoan-Indus Valley trade routes implied by common weight measures. In Proceedings of the 26th International Database Engineering amp; Applications Symposium (IDEAS 2022). ACM Press, New York, NY, USA

[13] Shambavi Sadayappan, Ian McCulloh, and John Piorkowski. 2018. Evaluation of Political Party Cohesion Using Exponential Random Graph Modeling. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '18). IEEE Press, Piscataway, NJ, 298-301.

[14] Evgenia Tsafou, and Juan José García-Granero, 2021. Beyond staple crops: Exploring the use of 'invisible' plant ingredients in Minoan cuisine through starch grain analysis on ceramic vessels. Archaeological and Anthropological Sciences, 13, 128. https://doi.org/10.1007/s12520-021-01375-4

[15] Bryan Wells and Andreas Fuls, 2017. Online Indus Writing Database. http://www.indus.epigraphica.de/

[16] Wikipedia, Proto-Uralic language, Available at: https://en.wikipedia.org/wiki/Proto-Uralic_language

# Author List

# Author List(Continued)

ISBN 978-1-4503-9709-4

15100 >

9 781450 397094